

Universitat de Lleida
Escola Politècnica Superior
Enginyeria Tècnica en Informàtica de Sistemes
Treball de final de carrera

Desarrollo de un cuadro de mando para una gran compañía

Autor: Jorge Zagalaz Lasierra
Directores: Josep Maria Pedrós Sentañes y Ramiro Moreno Chiral
Julio de 2008

Agradecimientos

Antes de empezar el proyecto en sí, me gustaría dedicar unas líneas a la gente que ha hecho posible que este proyecto vea la luz y a los amigos que he ido haciendo en este tiempo.

En primer lugar a mi familia, que han sido los que más me han apoyado e insistido para que finalizara la carrera y que tantas veces me han tenido que repetir esa frase de “Jorge estudia”.

Es de justicia, que el segundo en ser nombrado sea Ramiro, que me acogió como proyectista cuando no tenía tutor y no sabía con quien podría hacer el proyecto. Al final, el proyecto que iniciamos no ha podido llegar a su fin y hablando en clave cinematográfica, decidimos cambiar el guión siguiendo con los mismos protagonistas.

El tercer protagonista de esta memoria es Josep Maria, codirector del proyecto, responsable del mismo en Indra y buen tío en general. Apoyando mi propuesta desde el primer día que se lo comenté y animándome a ir terminando los plazos del proyecto en las fechas que habíamos marcado.

Si de Indra hablamos, también tenemos que nombrar a Roberto Raluy, jefe del área de SAP en la Software Lab de Indra en Lleida, quien también me ayudó a tomar la decisión de realizar el proyecto en Indra.

Por último, quiero agradecer a los amigos que he hecho en Indra, aunque algunos ya lo eran de la época universitaria. Teniendo a gente como Joan, Raúl, Javi, Gemma, Jaume, Víctor, Alberto, Alejandro, Gerardo, Albert o Martín uno va a trabajar contento porque conoce el buen ambiente que reina en la empresa y sabe que tiene compañeros que te echarán una mano en caso de necesidad.

Índice

1. Introducción	1
1.1. Indra	1
1.1.1. Historia de Indra	2
1.1.2. Estructura Accionarial	3
1.2. Situación actual y objetivos del proyecto	4
1.2.1. Descripción de la necesidad	4
1.2.2. Enfoque	4
1.2.3. Objetivos	5
1.3. Solución que proponemos	6
1.4. Perfil de usuarios	7
1.4.1. Tipos de usuarios:	7
1.5. Usabilidad	8
2. SAP	9
2.1. Planificación de Recursos Empresariales (ERP)	9
2.1.1. Los objetivos principales de los sistemas ERP son:	9
2.1.2. Ventajas de un ERP	11
2.1.3. Desventajas de los ERP	12
2.1.4. ERP's disponibles en el mercado	13
2.2. SAP	16
2.2.1. ¿Que es SAP?	16
2.2.2. Historia	16
2.2.3. Productos que ofrece SAP	17
2.2.4. ABAP	20
2.2.5. Módulos de SAP R/3	21
2.2.6. Mercados en los que trabaja SAP	25
3. BW	26
3.1. Inteligencia Empresarial	26
3.1.1. Historia	26
3.1.2. Características	27
3.2. SAP Business Information Warehouse (SAP BW)	28

4. Planificación temporal y valoración económica	30
4.1. Planificación temporal	30
4.2. Valoración económica	31
5. Diseño Funcional	33
5.1. Introducción y objetivos	33
5.2. Modelo de sistema	34
5.3. Modelo de datos	36
5.4. Interfaz de usuario	39
6. Tecnologías utilizadas	41
6.1. Lenguajes de programación usados	41
6.1.1. ABAP	41
6.1.2. HTML	42
6.1.3. JavaScript	42
6.2. Programas utilizados	44
6.2.1. Sap Logon	44
6.2.2. Query Designer	58
6.2.3. Web Application Designer (WAD)	76
6.2.4. Business Explorer Analyzer	82
7. Desarrollo e implementación	84
7.1. Pantallas de navegación	84
7.2. Pantallas de información	88
8. Conclusiones	105
9. Apéndices	107
9.1. La ayuda de la aplicación	107
9.2. Funciones javascript utilizadas	108
10. Bibliografía	120

Índice de figuras

1.	Accionistas Indra	3
2.	Esquema de los diferentes módulos de R/3	24
3.	Hitos en la planificación del proyecto	30
4.	Esquema temporal de la planificación	31
5.	Mapa de Sistemas	35
6.	Modelo de datos	36
7.	Datos maestros del modelo de datos	37
8.	Documento de formulación de las queries	39
9.	Documento de maquetación de las pantallas	40
10.	RSA1: Menú de InfoProvider	45
11.	RSA1: Crear reglas de actualización de un ODS	46
12.	RSA1: Crear reglas de actualización de un ODS - 2	47
13.	RSA1: Crear reglas de actualización de un ODS - 3	48
14.	RSA1: Menú de InfoFuentes	50
15.	RSA1: Asignar fuente de datos a una InfoFuente	50
16.	RSA1: Reglas de transferencia de una InfoFuente	51
17.	RSA1: Infopaquete asignado a una InfoFuente	52
18.	RSA1: Tipos de Sistemas Fuente	52
19.	SE11: Diccionario ABAP	53
20.	SE38: Modificar programa	54
21.	SE38: Modificar pantalla	54
22.	SE80: Browser	55
23.	RSPC: Ejemplo de cadena de carga	56
24.	SMICM: Invalidar Caché del servidor	57
25.	Query Designer: Crear query; seleccionar InfoProvider	59
26.	Query Designer: Propiedades del query	60
27.	Query Designer: Pantalla inicial	61
28.	Query Designer: Crear una estructura nueva	61
29.	Query Designer: Usar estructura existente	62

30.	Query Designer: Columna, crear una nueva selección	63
31.	Query Designer: Columna, selección nueva	63
32.	Query Designer: Trabajar con las dimensiones de la selección . . .	64
33.	Query Designer: Ámbito de valores	65
34.	Query Designer: Ocultar selección	66
35.	Query Designer: Fórmula sencilla	67
36.	Query Designer: Fórmula de suma entre dos selecciones	68
37.	Query Designer: Selección sin ratio disponible	69
38.	Query Designer: Diseño de excepciones	70
39.	Query Designer: Tabla final con excepciones	70
40.	Query Designer: Definición celda a celda	71
41.	Query Designer: Query de ejemplo finalizada	72
42.	Query Designer: Área de columnas de una query de gráfico	72
43.	Query Designer: Ejemplo de una query de gráfico	73
44.	Ejemplo query tabla	74
45.	Query tabla: Vista final	74
46.	Ejemplo query gráfico	75
47.	Query gráfico: Vista final	75
48.	WAD: Pantalla general	76
49.	WAD: Subpantalla Item de web	77
50.	WAD: Subpantalla Propiedades	77
51.	WAD: Pantalla principal, pestaña Resumen	79
52.	WAD: Propiedades del gráfico	80
53.	WAD: Elegir por que característica filtramos en un selector	81
54.	WAD: Tratar gráfico mixto	81
55.	BEx: Barra de herramientas	82
56.	Bex: Habilitar macros	82
57.	Bex: Abrir query	83
58.	Pantalla de Inicio	84
59.	Mapa del Sitio	85
60.	Pantalla Operaciones Mantenimiento	88

61.	Pantalla Operaciones Mantenimiento: Cabecera	89
62.	Pantalla Operaciones Mantenimiento: Filtro cabecera	89
63.	Pantalla Operaciones Mantenimiento: Tabla Atención Avisos . . .	91
64.	Query correspondiente a la tabla de Atención Avisos	91
65.	Pantalla Operaciones Mantenimiento: Gráfico Inspección Periódica	93
66.	Query correspondiente al gráfico de Inspección Periódica	94
67.	Selector del gráfico	97
68.	Selector del detalle	98
69.	Comentarios, carrusel y enlaces de pie de página	100
70.	Pantalla Operaciones - Lectura y Medida	102
71.	Cambio de columna visible en la tabla	103
72.	Help Maker	107

Índice de cuadros

1.	Principales ERP del mercado	13
2.	WAD: colores para diferenciar código HTML	78

1. Introducción

1.1. Indra

Indra es la multinacional de Tecnologías de la Información número 1 en España y una de las principales de Europa y Latinoamérica. Es la segunda compañía europea por capitalización bursátil de su sector y es una de las tres empresas españolas que más invierte en I+D. En 2007 sus ventas superaron los 2.167 M€, de los que un tercio procedió del mercado internacional. Cuenta con más de 23.000 profesionales y con clientes en más de 82 países.

Indra está organizada en seis mercados verticales: Defensa y Seguridad, Transporte y Tráfico, Energía e Industria, Telecomunicaciones y Media, Finanzas y Seguros y AA.PP y Sanidad, en las que año tras año ha ido desarrollando un entendimiento profundo del negocio y estableciendo una fuerte relación con los clientes. Estos conocimientos, unidos a la tecnología más avanzada, permiten a Indra crear una oferta diferencial potenciando el desarrollo de soluciones propias para cada segmento de mercado.

Indra ofrece a sus clientes una oferta completa y de valor que incluye desde la consultoría, el desarrollo de proyectos y la integración de sistemas y aplicaciones hasta el outsourcing de sistemas de información y de procesos de negocios. Esta oferta se estructura en dos segmentos principales: Soluciones y Servicios.

La oferta de soluciones incluye una amplia gama de sistemas, aplicaciones y componentes para la captación de datos e información, su tratamiento, transmisión y posterior presentación, básicamente enfocados al control y gestión de procesos complejos y/o críticos. Asimismo, Indra posee una extensa oferta de consultoría, que incluye consultoría tecnológica, de operaciones y estratégica, siendo ésta última suministrada por nuestra filial Europraxis.

En general, las soluciones de Indra sirven al núcleo de la operación del negocio de nuestros clientes y requieren como capacidad esencial la integración de sistemas, es decir, el diseño, configuración, desarrollo e implantación de componentes, aplicaciones y sistemas de información completos, incorporando productos propios y de terceros.

Mediante la oferta de servicios, Indra gestiona y explota sistemas y soluciones (Outsourcing, AM, Mantenimiento, etc...), así como determinados procesos de negocio donde la tecnología es un elemento estratégico y diferencial (BPO). La Gestión de Procesos de Negocio donde la tecnología es un elemento estratégico y diferencial, Indra la desarrolla a través de su filial Indra bmb. En Indra entendemos la externalización bajo una perspectiva de socio tecnológico, construyendo con nuestros clientes una estrecha relación donde el servicio se liga a la evolución de su negocio, compartiendo objetivos, riesgos y beneficios.

Indra constituye una referencia destacada en los mercados en que opera, tanto a escala nacional como internacional. Con referencias en más de 80 países de los

cinco continentes, de sus ingresos anuales, aproximadamente un tercio proceden de los mercados internacionales.

Algunas de las principales referencias de Indra son las siguientes:

- Un tercio del tráfico aéreo mundial está gestionado por países que utilizan los sistemas desarrollados por Indra para la gestión del tráfico aéreo.
- Algunos de los principales metros del mundo, como los de Madrid, Barcelona, París, Shanghai, Atenas o Santiago de Chile, entre otros, utilizan los más modernos sistemas de ticketing desarrollados por la compañía.
- Son muchos los países que han confiado en nosotros para sus procesos electorales; la red de defensa aérea del Estado español está desarrollada con tecnología de Indra y los simuladores de aviones que Indra desarrolla han sido calificados como los mejores del mundo por sus usuarios.
- Algunas de las principales compañías del mercado han confiado en Indra para sus procesos de desarrollo, integración y consultoría de sistemas, así como de outsourcing.
- Más de 120 compañías de utilities han implantado nuestras soluciones tecnológicas.

1.1.1. Historia de Indra

Indra es hoy la compañía española de referencia en Tecnologías de la Información, formada por más de 23.000 profesionales con un conocimiento profundo de la tecnología y del negocio de sus clientes. Hasta llegar aquí, la compañía ha vivido diferentes hitos que configuran su historia, paralela al desarrollo tecnológico de nuestro país.

El origen de las actividades de Indra se remonta a 1921, cuando se constituyó en la localidad madrileña de Aranjuez la primera de las empresas que, posteriormente y tras varias modificaciones de carácter patrimonial y societario, daría lugar a la configuración, en el año 1993, de lo que hoy es Indra.

Este año Indra englobaba los negocios de un amplio grupo de empresas, procedentes tanto del sector público como del sector privado, que concentraban gran parte de la oferta tradicional española en el campo de las Tecnologías de la Información. Es entonces cuando comienza la definición del actual proyecto empresarial.

Entre 1996 y 1999, Indra impulsó su proceso de consolidación y crecimiento. Durante este período la compañía afianzó su posición de liderazgo en el mercado español, con una atractiva presencia internacional, alcanzando niveles de rentabilidad muy favorablemente comparables con los de otras empresas internacionales del sector, que culminó con su salida a bolsa en marzo de 1999.

En el período 2000-2005 Indra experimentó un fuerte crecimiento por encima de la media del sector, gracias a un modelo de negocio diferencial y a una

mayor presencia en mercados internacionales, apoyada en la creación de filiales en países como EE.UU, China, Portugal y Brasil, entre otros.

En el año 2006 Indra integra con éxito los negocios de Azertia y Soluziona.

Azeria nace en 2001 como filial de tecnologías de la información de Corporación IBV (grupo empresarial al 50 % de Iberdrola y BBVA), fruto de la integración de Centrisa, Keon y TeleInformática. Posteriormente, AZERTIA fue creciendo con la incorporación de KRISTINA y SEINTEX, Euroquality, Socintec y Dimensión Informática.

Soluziona se remonta a los años ochenta, cuando Unión Fenosa creó una serie de empresas de servicios profesionales cuyo objetivo fue la puesta en valor de los conocimientos y experiencias de la compañía. En el año 2000 ese grupo de empresas se constituye como Soluziona, aportando el bagaje de veinte años de experiencia en el mercado nacional e internacional y un fuerte conocimiento en los sectores de Utilities y Telecomunicaciones, así como de otras áreas de negocio.

1.1.2. Estructura Accionarial

La Compañía no dispone de un registro nominal de sus accionistas, por lo que únicamente puede conocer la composición de su accionariado por la información que éstos le comuniquen directamente o hagan pública en aplicación de la normativa vigente sobre participaciones significativas (que obliga a comunicar, con carácter general, participaciones superiores al 3 % del capital), y por la información que facilita Iberclear, que la Sociedad recaba con ocasión de la celebración de sus juntas generales de accionistas.

De acuerdo con lo anterior, de la información conocida por la sociedad resulta que sus accionistas principales con participación superior al 5 % son: Unión Fenosa, con un 15 %; Caja Madrid, con un 14,83 %; Casa Grande de Cartagena, con un 5,68 %; y CajAstur con un 5 %.

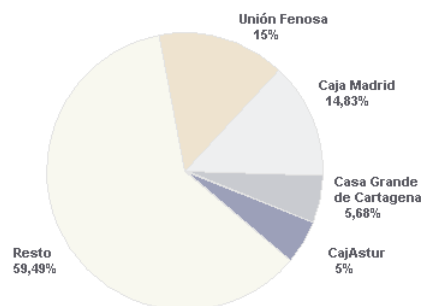


Figura 1: Accionistas Indra

1.2. Situación actual y objetivos del proyecto

Una gran compañía española, vino a nosotros, Indra, para que desarrolláramos un cuadro de mando para una de sus secciones. El tema en cuestión era el de construir e implantar el cuadro de mando operativo de negocios regulados y el posterior mantenimiento del mismo.

Antes de continuar, definiremos el concepto de cuadro de mando sobre el que se basará el proyecto. Un cuadro de mando es una herramienta de administración de empresas que muestra continuamente cuándo una compañía y sus empleados alcanzan los resultados definidos por el plan estratégico. Proporciona a los administradores una mirada global de las prestaciones del negocio y ayuda a la compañía a expresar los objetivos e iniciativas necesarias para cumplir con la estrategia.

1.2.1. Descripción de la necesidad

Nos encontramos en la etapa final de liberalización del mercado que engloba a nuestro cliente, situación que genera un escenario a corto plazo con un alto grado de incertidumbre a nivel de negocio, que tiene repercusiones, tanto en la segregación de funciones existente en la actualidad, como en los diferentes esquemas retributivos del Grupo.

En este entorno regulatorio, nuestro cliente precisa de herramientas que faciliten la toma de decisiones y agilicen los procedimientos de gestión interna.

Sin embargo, estos cambios a nivel de negocio sólo afectarán el cuadro de mando en la capa de agregación. Por esta razón es necesario estructurar esta herramienta de manera que permita tanto hacer un seguimiento de la actividad como la gestión de cambio del modelo.

Por tanto, el Cuadro de mando (CM) ha de tener como objetivos:

Seguir el desarrollo de los negocios

Ofrecer los ratios básicos por negocio

Que permita a cada unidad fijar y seguir unos objetivos de negocio sobre los que se pueda actuar

Evaluar las decisiones estratégicas en los cambios de modelo

Evaluar y simular los diferentes impactos.

1.2.2. Enfoque

El Cuadro de Mando de negocios regulados (CMNNRR) deberá considerar una serie de premisas fundamentales y deberá ajustarse a la realidad de los usuarios.

Deberá proporcionar las dimensiones de análisis necesarias, con las perspectivas de negocio requeridas y que de un enfoque cualitativo dotando a la herramienta para que dé soporte a la toma de decisiones y facilite la consecución de los objetivos del negocio.

1.2.3. Objetivos

El CMNNRR tendrá como objetivo principal:

Dotar a la Dirección General de Negocios Regulados de una herramienta que dé soporte a la toma de decisiones y facilite la consecución de sus objetivos de negocio

Y como objetivos secundarios:

Enlazar la información de gestión con la que se entrega al Consejo de Administración a través del Cuadro de Mando Corporativo.

Agilizar los plazos de obtención y preparación, así como facilitar el análisis de la información por parte de los usuarios finales y del departamento de Control de Gestión.

Homogeneizar la terminología utilizada en la compañía.

Enriquecer, a través del análisis multidimensional, la información de gestión del Área.

Siguiendo con la actual filosofía del Cuadro de Mando Corporativo, a la perspectiva financiera se incorporan las perspectivas de operación y de calidad con los siguientes objetivos:

Integrar las tres líneas estratégicas del Grupo (Crecimiento, Rentabilidad y Calidad) en un único entorno.

Facilitar la información relevante a los usuarios de forma que se cubran todas sus necesidades de gestión.

1.3. Solución que proponemos

Tras escuchar al cliente y ver que la aplicación que nos pedían era para ayudar en la toma de decisiones importantes dentro de la empresa, la solución, era usar Business Intelligence (BI), que es el conjunto de estrategias y herramientas enfocadas a la administración y creación de conocimiento mediante el análisis de datos existentes en una organización o empresa.

Como todas las aplicaciones de nuestro cliente estaban implementadas con el ERP, SAP, decidimos implementar este proyecto con la sección de inteligencia empresarial de la compañía alemana, llamada SAP Business Information Warehouse (SAP BW) en la cual profundizaremos más adelante.

Tras varias entrevistas en profundidad con los responsables de las áreas implicadas del cliente, realizamos el diseño conceptual, se hizo una maquetación del modelo de datos y de las pantallas que debía tener la aplicación sobre la cual nos hemos basado en el desarrollo del proyecto.

A lo largo del proyecto, iremos reuniéndonos con el cliente, para mostrar entregas parciales, evaluarlas y mejorar lo que veamos incorrecto.

1.4. Perfil de usuarios

Es importante la definición de los perfiles de usuario que accederán al CM-NRRR para preparar las autorizaciones. Se debe tener en cuenta que el nivel de autorización será por pantalla.

En un principio, esta aplicación será utilizada por altos mandos de la empresa que nos ha contratado, y que servirá como apoyo a la toma de decisiones sobre el rumbo de la empresa.

Asimismo tenemos en cuenta que los usuarios finales tienen unos conocimientos básicos sobre el uso de Internet, puesto que la navegación en nuestra aplicación será similar a la de una página Web.

1.4.1. Tipos de usuarios:

Desarrolladores:

- Jefe de proyecto: es el encargado de reunirse con el cliente para hacer un diseño inicial y negociar posibles cambios mientras se desarrolla el proyecto. A estas reuniones también acude el analista para tratar la parte técnica.
- Analista: conociendo los deseos del cliente y la posibilidad de llevarlos a cabo de una manera eficaz en el tiempo previsto, desarrolla el diseño técnico en el que se basaran los programadores para dar forma al proyecto.
- Programador: desarrollan la aplicación en función del diseño técnico que ha creado el analista.

Usuarios finales

- Súper usuario: puede acceder a todas las secciones de la aplicación sin restricción alguna.
- Directivos de sección: sólo puede ver lo correspondiente a la sección que dirige, puesto que no le debería interesar la información de otras secciones de la empresa para su trabajo diario.
- Directivos que controlan todas las secciones: pueden acceder a todas las pantallas pero en modo lectura.

1.5. Usabilidad

Los objetivos básicos de la usabilidad los podemos enumerar en la siguiente lista:

- **Facilidad en el aprendizaje:** la aplicación es muy sencilla de utilizar, por consiguiente el aprendizaje debería ser rápido y sin inconvenientes si los usuarios tienen unos mínimos conocimientos de navegación Web como ya hemos comentado antes al definir los posibles usuarios de la aplicación.
- **Consistencia:** la aplicación cumple este objetivo, representa en los gráficos los datos obtenidos en las tablas, puesto que es una de sus funciones. También los datos de las tablas de detalle, son los mismos que obtenemos en las tablas principales.
- **Flexibilidad:** nuestra aplicación solo tendrá flexibilidad en las fechas que abarcaran los datos a mostrar y en el gráfico que queramos mostrar, puesto que no podremos introducir o modificar los datos que tenemos.
- **Robustez:** una vez entregado el proyecto al cliente, una de sus principales virtudes es la robustez, ya que hemos tenemos estabilidad y respuesta en las pruebas o consultas que queramos realizar.
- **Recuperabilidad:** la aplicación permite recuperar datos de fechas anteriores. Para volver a mostrar una consulta realizada anteriormente tenemos que incluir los mismos parámetros que en la anterior
- **Tiempo de respuesta:** actualmente, la aplicación no es tan rápida como desearíamos, esto es debido a las consultas que tenemos que realizar para cargar algunas pantallas. Es uno de los temas a mejorar en posibles actualizaciones del proyecto.
- **Adecuación de tareas:** como hemos comentado, la aplicación se usa como apoyo a la toma de decisiones, no sirve para tomarlas, por tanto la adecuación de tareas no estaría entre las características de usabilidad de este proyecto.
- **Disminución de la carga cognitiva:** la sencillez de las pantallas que tendrán ante si los usuarios no implica una gran carga cognitiva. Es más, están hechas con la intención de que esta carga sea mínima.

2. SAP

Como hemos comentado en el primer capítulo, para solucionar el problema de nuestro cliente, usaremos el ERP de SAP, R/3.

Para empezar explicaremos que es un ERP y los existentes en el mercado. Posteriormente, profundizaremos en SAP, hablando de los módulos en los cuales está dividido y los servicios que ofrece a las empresas.

2.1. Planificación de Recursos Empresariales (ERP)

Los sistemas de Planificación de Recursos Empresariales, ERP, (Enterprise resource planning) son sistemas de gestión de información, que integran y automatizan la mayoría de las prácticas de negocio asociadas con los aspectos operativos o productivos de una empresa comprometida en la producción de bienes o servicios.

La Planificación de Recursos Empresariales es un término derivado de la Planificación de Recursos de Manufactura (MRPII) y seguido de la Planificación de Requerimientos de Material (MRP). Los sistemas ERP típicamente manejan la producción, logística, distribución, inventario, envíos, facturas y una contabilidad para la compañía de la Planificación de Recursos Empresariales o el software ERP puede intervenir en el control de muchas actividades de negocios como ventas, entregas, pagos, producción, administración de inventarios, calidad de administración y la administración de recursos humanos.

Los sistemas ERP son sistemas integrales de gestión para la empresa. Se caracterizan por estar compuestos por diferentes partes integradas en una única aplicación. Estas partes son de diferente uso, por ejemplo: producción, ventas, compras, logística, contabilidad (de varios tipos), gestión de proyectos, GIS (sistema de información geográfica), inventarios y control de almacenes, pedidos, nóminas, etc. Sólo podemos definir un ERP como la integración de todas estas partes. Lo contrario sería como considerar un simple programa de facturación como un ERP por el simple hecho de que una empresa integre únicamente esa parte. Ésta es la diferencia fundamental entre un ERP y otra aplicación de gestión. El ERP integra todo lo necesario para el funcionamiento de los procesos de negocio de la empresa. No podemos hablar de ERP en el momento que tan sólo se integra uno o una pequeña parte de los procesos de negocio. La propia definición de ERP indica la necesidad de "Disponibilidad de toda la información para todo el mundo todo el tiempo".

2.1.1. Los objetivos principales de los sistemas ERP son:

- Optimización de los procesos empresariales.
- Acceso a toda la información de forma confiable, precisa y oportuna (integridad de datos).

- La posibilidad de compartir información entre todos los componentes de la organización.
- Eliminación de datos y operaciones innecesarias de reingeniería.

El propósito fundamental de un ERP es otorgar apoyo a los clientes del negocio, tiempos rápidos de respuesta a sus problemas, así como un eficiente manejo de información que permita la toma oportuna de decisiones y disminución de los costos totales de operación.

Las características que distinguen a un ERP de cualquier otro software empresarial, es que deben de ser sistemas integrales, con modularidad y adaptables:

- Integrales, porque permiten controlar los diferentes procesos de la compañía entendiendo que todos los departamentos de una empresa se relacionan entre sí, es decir, que el resultado de un proceso es punto de inicio del siguiente. Por ejemplo, en una compañía, el que un cliente haga un pedido representa que se cree una orden de venta que desencadena el proceso de producción, de control de inventarios, de planificación de distribución del producto, cobranza, y por supuesto sus respectivos movimientos contables. Si la empresa no usa un ERP, necesitará tener varios programas que controlen todos los procesos mencionados, con la desventaja de que al no estar integrados, la información se duplica, crece el margen de contaminación en la información (sobre todo por errores de captura) y se crea un escenario favorable para malversaciones. Con un ERP, el operador simplemente captura el pedido y el sistema se encarga de todo lo demás, por lo que la información no se manipula y se encuentra protegida.
- Modulares. Los ERP entienden que una empresa es un conjunto de departamentos que se encuentran interrelacionados por la información que comparten y que se genera a partir de sus procesos. Una ventaja de los ERP, tanto económica como técnica es que la funcionalidad se encuentra dividida en módulos, los cuales pueden instalarse de acuerdo con los requerimientos del cliente. Ejemplo: ventas, materiales, finanzas, control de almacén, recursos humanos, etc.
- Adaptables. Los ERP están creados para adaptarse a la idiosincrasia de cada empresa. Esto se logra por medio de la configuración o parametrización de los procesos de acuerdo con las salidas que se necesiten de cada uno. Por ejemplo, para controlar inventarios, es posible que una empresa necesite manejar la partición de lotes pero otra empresa no. Los ERP más avanzados suelen incorporar herramientas de programación de 4^a Generación para el desarrollo rápido de nuevos procesos. La parametrización es el valor añadido fundamental que debe contar cualquier ERP para adaptarlo a las necesidades concretas de cada empresa.

Otras características destacables de los sistemas ERP son:

- Base de datos centralizada.
- Los componentes del ERP interactúan entre sí consolidando todas las operaciones.
- En un sistema ERP los datos se ingresan sólo una vez y deben ser consistentes, completos y comunes.
- Las empresas que lo implanten suelen tener que modificar alguno de sus procesos para alinearlos con los del sistema ERP. Este proceso se conoce como Reingeniería de Procesos, aunque no siempre es necesario.
- Aunque el ERP pueda tener menús modulares configurables según los roles de cada usuario, es un todo. Esto significa: es un único programa (con multiplicidad de bibliotecas, eso sí) con acceso a una base de datos centralizada. No debemos confundir en este punto la definición de un ERP con la de una suite de gestión.
- La tendencia actual es a ofrecer aplicaciones especializadas para determinadas empresas. Es lo que se denomina versiones sectoriales o aplicaciones sectoriales especialmente indicadas o preparadas para determinados procesos de negocio de un sector (los más utilizados).

Las soluciones ERP en ocasiones son complejas y difíciles de implantar debido a que necesitan un desarrollo personalizado para cada empresa partiendo de la parametrización inicial de la aplicación que es común. Las personalizaciones y desarrollos particulares para cada empresa requieren de un gran esfuerzo en tiempo, y por consiguiente en dinero, para modelar todos los procesos de negocio de la vida real en la aplicación.

Las metodologías de implantación de los ERP en la empresa no siempre son todo lo simples que se desearía, dado que entran en juego múltiples facetas.

2.1.2. Ventajas de un ERP

Un fabricante que no disponga de un ERP, en función de sus necesidades, puede encontrarse con muchas aplicaciones de software cerradas, que no se pueden personalizar, y no se optimizan para su negocio. Diseño de ingeniería para mejorar el producto, seguimiento del cliente desde la aceptación hasta la satisfacción completa, una compleja administración de interdependencias de los recibos de materiales, de los productos estructurados en el mundo real, de los cambios de la ingeniería y de la revisión y la mejora, y la necesidad de elaborar materiales sustitutos, etc. La ventaja de tener un ERP es que todo esto, y más, está integrado.

La seguridad de los ordenadores está incluida dentro del ERP, para proteger en contra de ataques externos, tal como el espionaje industrial y ataques internos, como puede ser la malversación. Una falsificación en el escenario de los datos

puede involucrar terrorismo alterando el recibo de. La seguridad del ERP ayuda a prevenir el abuso.

2.1.3. Desventajas de los ERP

Muchos de los problemas que tienen las compañías con el ERP son debido a la inversión inadecuada para la educación continua del personal relevante, incluyendo los cambios de implementación y de prueba, y una falta de políticas corporativas que afectan a cómo se obtienen los datos del ERP y como se mantienen actualizados.

Algunas de las desventajas que nos encontramos con los ERP son las siguientes:

- El éxito depende en las habilidades y la experiencia de la fuerza de trabajo, incluyendo la educación y como hacer que el sistema trabaje correctamente. Muchas compañías no invierten lo suficiente en formación. Los propietarios de pequeñas empresas están menos capacitados, lo que significa que el manejo del sistema ERP es operado por personal que no está capacitado para el manejo del mismo.
- Cambio de personal, las compañías pueden emplear administradores que no están capacitados para el manejo del sistema ERP, proponiendo cambios en las prácticas de los negocios que no están sincronizados con el sistema.
- La instalación del sistema ERP es muy costosa.
- Los vendedores del ERP pueden cargar sumas de dinero para la renovación de sus licencias anuales o para realizar una migración a una versión más reciente.
- Los ERP son vistos como sistemas muy rígidos, y difíciles de adaptarse al flujo específico de los trabajadores y el proceso de negocios de algunas compañías.
- Una vez que el sistema esté establecido, los costos de los cambios son muy altos (reduciendo la flexibilidad y las estrategias de control).
- La resistencia en compartir la información interna entre departamentos puede reducir la eficiencia del software.
- Los sistemas pueden tener excesiva ingeniería respecto a las necesidades reales del consumidor.

2.1.4. ERP's disponibles en el mercado

En el mercado disponemos de ERP de software libre y ERP de los que necesitamos obtener una licencia del propietario para usarlos.

En el 2005, el mercado de la planificación de recursos empresariales estaba dominado por SAP y de distribuía de la siguiente manera:

	ERP	GANANCIAS (M€)	CUOTA DE MERCADO (%)
1	SAP	4726	28.7
2	Oracle Applications	1674	10.2
3	The Sage Group	1221	7.4
4	Microsoft Dynamics	616	3.7
5	SSA Global Technologies	464	2.8

Cuadro 1: Principales ERP del mercado

2.1.4.1. ERP de software libre

- Adempiere
- Compiere
- ERP5
- GNU Enterprise
- JFire
- Kuali Foundation
- LedgerSMB
- OFBiz
- OpenBlueLab
- Openbravo
- Opentaps
- Postbooks
- Tiny ERP
- SQL-Ledger
- WebERP

2.1.4.2. ERP con licencia

- 1C:Enterprise from 1C Company
- 24SevenOffice Start, Premium, Professional and Custom from 24SevenOffice
- abas ERP from ABAS Software
- Accpac from The Sage Group
- Agresso Business World from Unit 4 Agresso
- AMS Advantage from CGI Group (formerly American Management Systems)
- BatchMaster ERP from BatchMaster Software
- Comprehensive Patient Administrator
- Enterprise Business System from Made2Manage Systems
- Epicor Enterprise from Epicor
- ERP Adage (aka Adage) from Infor Global Solutions
- ERP LN (aka Baan) from Infor Global Solutions
- ERP LX (aka BPCS) from Infor Global Solutions
- ERP SL (aka SyteLine) from Infor Global Solutions
- ERP XA (aka MAPICS) from Infor Global Solutions
- Global Shop Solutions One-System ERP Solutions
- HansaWorld products
- IFS Applications from Industrial and Financial Systems
- JD Edwards EnterpriseOne & JD Edwards World from Oracle
- kVASy4 from SIV.AG
- Lawson Financials from Lawson Software
- Maximo (MRO) from IBM
- MFG/PRO from QAD
- Microsoft Dynamics from Microsoft
- Momentum from CGI Group

- Movex from Intenia
- NetERP from NetSuite Inc.
- Openda QX from Openda
- OpenMFG from xTuple
- Oracle e-Business Suite from Oracle
- Paradigm from Consona Corporation
- PeopleSoft from Oracle
- Ramco e.Applications from Ramco Systems
- MAS 90, MAS 200 and MAS 500 from The Sage Group
- SAGE ERP X3 from The Sage Group
- SAP R/3 from SAP
- SYSPRO from Syspro
- mySAP from SAP
- Visual Enterprise from Infor Global Solutions

2.2. SAP

Como hemos visto en la página anterior disponemos de un amplio abanico de posibilidades a la hora de elegir que ERP usar, pero como hemos comentado en el primer capítulo, utilizaremos SAP, porque es el sistema que usa nuestro cliente y de este modo la aplicación que implementemos en este proyecto tendrá un mejor acoplamiento al sistema con el que trabaja actualmente nuestro cliente.

2.2.1. ¿Que es SAP?

SAP AG (Systeme, Anwendungen und Produkte) (Sistemas, Aplicaciones y Productos), con sede en Walldorf (Alemania), es el primer proveedor de software empresarial en el mundo. Como empresa, comercializa un conjunto de aplicaciones de software para soluciones integradas de negocios, entre ellas mySAP Business Suite, que provee soluciones escalables que permiten mejorar continuamente, con más de 1.000 procesos de negocio consideradas las mejores prácticas empresariales.

SAP es considerada como el tercer proveedor independiente de software del mundo y el mayor fabricante europeo de software. Con 12 millones de usuarios, 100.700 instalaciones, y más de 1.500 socios, es la compañía más grande de software Inter-empresa. A finales de 2005, SAP empleaba a 35.873 personas (fuente empleados) en más de 50 países y sus ingresos anuales fueron de 8.513 millones de euros (fuente ingresos).

2.2.2. Historia

SAP fue fundada en 1972 en la Ciudad de Mannheim, Alemania, por antiguos empleados de IBM (Claus Wellenreuther, Hans-Werner Hector, Klaus Tschira, Dietmar Hopp y Hasso Plattner) bajo el nombre de "SAP Systemanalyse, Anwendungen und Programmentwicklung". El nombre fue tomado de la división en la que trabajaban en IBM.

La corporación se ha desarrollado hasta convertirse en la quinta más grande compañía mundial de software. El nombre SAP R/3 es al mismo tiempo el nombre de una empresa y el de un sistema informático. Este sistema comprende muchos módulos completamente integrados, que abarca prácticamente todos los aspectos de la administración empresarial. Ha sido desarrollado para cumplir con las necesidades crecientes de las organizaciones mundiales y su importancia esta más allá de toda duda. SAP ha puesto su mirada en el negocio como un todo, así ofrece un sistema único que soporta prácticamente todas las áreas en una escala global. SAP proporciona la oportunidad de sustituir un gran número de sistemas independientes, que se han desarrollado e instalado en organizaciones ya establecidas, con un solo sistema modular. Cada módulo realiza una

función diferente, pero está diseñado para trabajar con otros módulos. Está totalmente integrado ofreciendo real compatibilidad a lo largo de las funciones de una empresa.

Después de haber dominado el mercado, la empresa afronta una mayor competencia de Microsoft e IBM. En marzo de 2004 cambió su enfoque de negocio en favor de crear la "plataforma" que desarrolla y utiliza, la nueva versión de su software NetWeaver.

Es en este punto donde SAP se encuentra enfrentado con Microsoft e IBM, en lo que se conoce como "la guerra de las plataformas". Microsoft ha desarrollado una plataforma basada en la Web llamada .NET, mientras IBM ha desarrollado otra llamada WebSphere.

A comienzos de 2004 sostuvo conversaciones con Microsoft sobre una posible fusión. Las empresas dijeron que las conversaciones finalizaron sin un acuerdo. Sin embargo, a comienzos del 2006 fue anunciada una alianza muy importante entre SAP y Microsoft para integrar las aplicaciones ERP de SAP con las de Office de Microsoft bajo el nombre de proyecto "Duet".

La compra de SAP por parte de Microsoft habría sido uno de los acuerdos más grandes en la historia de la industria del software, dado el valor de mercado de la alemana, de más de 55.000 millones de euros (junio 2004).

SAP ha conquistado clientes de forma consistente para aumentar la cuota del mercado global entre sus cuatro principales competidores a un 55 % a fines de 2004, desde un 48 % dos años antes. La participación combinada de Oracle y PeopleSoft declinó de un 29 % a un 23 %.

SAP es una compañía alemana, pero opera en todo el mundo, con 28 sucursales y afiliadas y 6 compañías asociadas, manteniendo oficinas en 40 países.

2.2.3. Productos que ofrece SAP

SAP trabaja en el sector de software de planificación de recursos empresariales (ERP). El principal producto de la compañía es el software SAP ERP, llamado hasta mediados de 2007 como SAP R/3, en el que la R significa procesamiento en tiempo real y el número 3 se refiere a las tres capas de la arquitectura de proceso: bases de datos, servidor de aplicaciones y cliente. El predecesor de R/3 fue R/2.

Otros productos de SAP son APO (Advanced Planner and Optimizer), BW (Business Information Warehouse), BI (Business Intelligence), Customer Relationship Management (CRM), SRM (Supplier Relationship Management), Human Resource Management Systems (EHRMS), Product Lifecycle Management (PLM), KW (Knowledge Warehouse) RE (Real Estate), FI/CO (Financial Accounting/Controlling).

SAP también ofrece una nueva plataforma tecnológica conocida como SAP NetWeaver. Esta plataforma tecnológica convierte a SAP en un programa Web-enabled, lo que significa que estaría totalmente preparado para trabajar con él mediante la web, se puede trabajar con SAP mediante cualquier navegador de internet si se tienen los componentes apropiados de SAP NetWeaver (SAP Portals).

Aunque sus principales aplicaciones están destinadas a grandes empresas, SAP también se dirige a la pequeña y mediana empresa con productos como SAP Business One y mySAP All-in-one.

SAP cuenta también con verticales y microverticales. Las verticales son conocidas también como IS o Industry Solution y son SAP orientados a diversas industrias, como por ejemplo periódicos, mineras, cías. de telecomunicaciones. Las microverticales son SAP que atienden a industrias específicas, como por ejemplo: empresas agroexportadoras, piscifactorías, etc. Las Verticales son desarrolladas por SAP y las microverticales por los socios de SAP.

En muchos casos la adopción de SAP por las empresas se hace mediante la contratación de consultoras especializadas.

SAP hoy en día ha logrado excelentes resultados en el mundo de los negocios.

2.2.3.1. R/3 El R/3 es el ERP creado por SAP. Es un sistema integrado de gestión que permite controlar todos los procesos que se llevan a cabo en un empresa, a través de módulos, los cuales detallaremos más adelante.

El sistema R/3 opera utilizando el principio cliente/servidor aplicado a varios niveles. Es altamente modular y se aplica fundamentalmente por medio del software, de forma que los modos de interacción entre los diversos clientes y servidores puedan ser controlados.

R/3 es compatible con las siguientes bases de datos:

- Informix
- Oracle
- Adabas
- Sybase ASE
- IBM DB/2
- Microsoft SQL Server

Y con los siguientes sistemas operativos:

- HP-UX

- AIX
- Citrix
- LINUX - Solo la versión comercial de Redhat
- Open VMS
- MPE/iX
- Windows Server
- IBM OS/400

2.2.3.2. NetWeaver Es una plataforma de tecnología integrada para todas las aplicaciones SAP en el plano técnico. Es conocida como una aplicación orientada a servicios y a la integración. SAP NetWeaver provee al usuario de un vínculo entre lenguajes y aplicaciones. Está construido usando estándares abiertos de la industria por lo que es sencillo negociar transacciones de información con desarrollos de Microsoft .NET, Sun Java EE, y IBM WebSphere.

NetWeaver fue lanzado como un movimiento estratégico de SAP que plantea a las empresas ejecutar todas sus aplicaciones empresariales en una única plataforma integrada con la más firme infraestructura. Esta solución incorpora un bajo costo con una gran flexibilidad, una mejor integración con las aplicaciones, construcción en estándares para asegurar la futura interoperación.

Este lanzamiento en suma es una parte del plan de SAP de transformarse en una herramienta más abierta y orientada a servicios en base a las necesidades del mercado.

2.2.3.3. BW El capítulo siguiente trata en profundidad la inteligencia empresarial, por consiguiente en este apartado simplemente nos limitamos a nombrarlo como un producto más de SAP.

2.2.4. ABAP

ABAP (Advanced Business Application Programming) es un lenguaje de cuarta generación, propiedad de SAP, que se utiliza para programar la mayoría de sus productos (R/3, mySAP Business suite...).

En el capítulo 6, en el apartado de lenguajes de programación usados en el desarrollo del proyecto, ya nos extenderemos más en este lenguaje.

2.2.5. Módulos de SAP R/3

El Sistema SAP R/3 consta, en la vista modular, de áreas empresariales homogéneas, que soportan las operaciones empresariales de una empresa y trabajan integradas en tiempo real.

La integración en SAP se logra a través de la puesta en común de la información de cada uno de los módulos y por la alimentación de una base de datos común. El sistema SAP está compuesto de una serie de módulos funcionales que responden de forma completa a los procesos operativos de las compañías.

Los módulos en los que está compuesto SAP son los siguientes:

Gestión Financiera (FI):

FI-GL	Cuentas de Mayor
FI-LC	Consolidación Sociedades
FI-AR	Cuentas a Cobrar
FI-AP	Cuentas a Pagar
FI-AA	Gestión de Activos
FI-SL	Special Ledger
Cierres	

Controlling (CO):

CO-CCA	Contabilidad por Centros Coste Contabilidad Presupuestaria
CO-PC	Control de Costes del Producto
CO-PA	Análisis de Rentabilidad
CO-OPA	Ordenes Internas
CO-ABC	Costes Basados en Actividades

Tesorería (TR):

Programa	Conciliación
Provisiones	Posicionamientos
Control	de Fondos

Sistema de proyectos (PS):

PS-BD	Datos Básicos
PS-OS	Planificación del proyecto
PS-PLN	Plan de Costes
PS-APM	Proceso de Aprobación
PS-EXE	Seguimiento y Progreso del Proyecto
PS-IS	Sistema de Información

Gestión de personal (HR):

HR-PA-EMP	Datos Maestros de Personal
HR-PA-PAY	Nómina
HR-PA-TRV	Gastos de Viaje
HR-PD-OM	Organización y Planificación
HR-PD-PD	Desarrollo de Personal
HR-PD-SCM	Gestión de la Formación
HR-PA-APP	Selección de Personal
HR-PA-TIM	Gestión de Tiempos

Mantenimiento (PM):

PM-EQM	Identificación Descripción
PM-PRM	Mantenimiento Preventivo
PM-WOC	Órdenes de Mantenimiento
PM-PRO	Proyectos de Mantenimiento
PM-SM	Gestión del Servicio

Gestión de calidad (QM):

QM-PT	Herramientas de planificación
QM-IM	Proceso de Inspección
QM-QC	Control de Calidad
QM-CA	Certificados de Calidad
QM-QN	Notificaciones de Calidad

Planificación de producto (PP):

PP-BD	Datos Básicos
PP-SOP	Gestión de la Demanda
PP-MP	Plan Maestro
PP-CRP	Plan de Capacidades
PP-MRP	Plan de Materiales
PP-SFC	Órdenes de Fabricación
PP-PC	Costes de producto
PP-IS	Sistema de Información
PP-PI	Industria de procesos
PP-CFG	Configuración de Producto

Gestión de material (MM):

MM-MRP	Planificación Necesidades Materiales
MM-PUR	Gestión de Compras
MM-IM	Gestión de Inventarios
MM-WM	Gestión de Almacenes
MM-IV	Verificación de Facturas
MM-IS	Sistema de Información
MM-EDI	Intercambio Electrónico de Datos
Sistema Clasificación	
Gestión de Lotes	

Comercial (SD):

SD-MD	Datos maestros
SD-SLS	Gestión de Ventas
SD-GF	Gestión Tarifas y Condiciones de Precio
SD-SHP	Gestión de Expediciones
SD-BIL	Facturación
SD-IS	Sistemas de Información
SD-EDI	Intercambio Electrónico de Datos

Workflow (WF).

Soluciones sectoriales (IS):

IS-R	Planificación de Surtidos
IS-R	Reaprovisionamiento
IS-R	Formatos de presentación
IS-R	Sales Retail
CP	Inventario de proveedores
MM	Compras Retail
SD	Transporte
RIS	Sistema de Información Retail

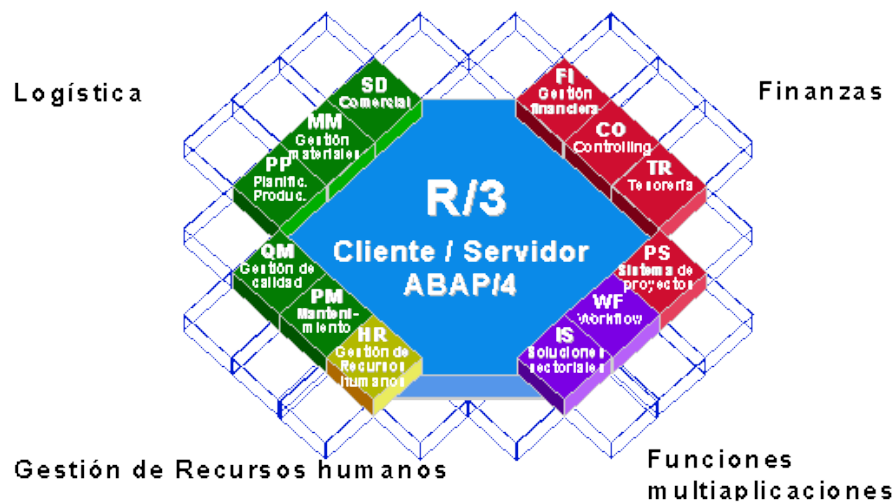


Figura 2: Esquema de los diferentes módulos de R/3

2.2.6. Mercados en los que trabaja SAP

SAP comercializa todos sus productos en diferentes industrias del mundo desde las compañías privadas hasta las gubernamentales. La siguiente lista es una muestra parcial de las industrias a las que sirve SAP:

- Materias primas, minería y agricultura
- Gas y petróleo
- Químicas
- Metalúrgicas
- Farmacéuticas
- Materiales de construcción, arcilla y vidrio
- Construcción pesada
- Servicios
- Consultorías y software
- Sanatorios y hospitales
- Muebles
- Automoción
- Textil y vestidos
- Papel y maderas
- Sector Público
- Retail
- Educación Superior e Investigación

3. BW

3.1. Inteligencia Empresarial

Es el conjunto de estrategias y herramientas enfocadas a la administración y creación de conocimiento mediante el análisis de datos existentes en una organización o empresa.

El propósito de la inteligencia empresarial es el de respaldar las decisiones tomadas por la empresa.

Los sistemas de inteligencia empresarial (BI, Business Intelligence) proveen visualizaciones históricas, en curso y predictivas de las operaciones de la empresa, usando datos que han sido recogidos en un almacén de datos y trabajando ocasionalmente con los datos de operaciones. Para comparar algunos datos, se toma información de otras empresas de la misma industria o sector, esto es conocido como benchmarking.

3.1.1. Historia

A principios del siglo XX, antes de que comenzara la edad de la información, las empresas tomaban los datos de fuentes no automatizadas, las empresas carecían de los recursos que actualmente tenemos para procesar los datos, y tomaban las decisiones sobre el rumbo que iba a tomar su negocio por intuición o basándose en lo que habían hecho las compañías de su mismo sector.

Cuando el mundo empresarial se empezó a automatizar, los datos comenzaron a almacenarse. Sin embargo, la recolección era un problema por la falta de medios e infraestructura para el intercambio de datos o a las incompatibilidades entre sistemas. Pese a todo, las empresas comenzaban a tener informes sobre sus datos, que les permitían tomar decisiones solamente a largo plazo debido a la tardanza de obtener dichos informes. Aun así, la intuición seguía estando presente en las decisiones a corto plazo.

Con el paso del tiempo, las empresas se fueron modernizando en tecnologías e incrementando la automatización de sus procesos. Se crearon bases de datos para almacenar todos sus resultados. El proceso de Extracción, transformación y carga (ETL) mejoró, y las herramientas para la integración de las aplicaciones en la empresa, fueron aumentando su capacidad para coleccionar datos.

La inteligencia empresarial ha conseguido separar entre las grandes cantidades de datos que disponía, los necesarios, extraer la información pertinente y convertir esta información en conocimientos de los que se pueden tomar decisiones para el futuro de la empresa.

En aplicaciones modernas del software de inteligencia empresarial, los directivos pueden compilar informes de los datos para pronosticar, el análisis, y la toma de decisiones de la empresa rápidamente.

Las señales interesantes para la inteligencia empresarial son las adquisiciones recientes por Oracle, SAP, IBM y Hewlett Packard (HP) en el área de la inteligencia de la empresa. IBM por ejemplo compró a un distribuidor de ECM líder del mercado, antes, Oracle compró Hyperion y HP compró Knightsbridge, un líder en la industria de BI.

Esto ha desembocado en una racionalización en el mercado y en el surgimiento de distribuidores que venden ERP, BI, transporte y manejo de datos y las tecnologías de Integración.

3.1.2. Características

Este conjunto de herramientas y metodologías tienen en común las siguientes características:

- Accesibilidad a la información. Los datos son la fuente principal de este concepto. Lo primero que deben garantizar este tipo de herramientas y técnicas será el acceso de los usuarios a los datos con independencia de la procedencia de estos.
- Apoyo en la toma de decisiones. Se busca ir más allá en la presentación de la información, de manera que los usuarios tengan acceso a herramientas de análisis que les permitan seleccionar y manipular sólo aquellos datos que les interesen.
- Orientación al usuario final. Se busca independencia entre los conocimientos técnicos de los usuarios y su capacidad para utilizar estas herramientas.

3.2. SAP Business Information Warehouse (SAP BW)

Sap Business Information Warehouse (SAP BW) es el nombre de la solución de inteligencia empresarial, análisis, creación de informes y almacenamiento de datos que nos ofrece SAP.

SAP BW consta entre otros elementos de almacén de datos, capacidad de modelado extensivo de datos, motor analítico, herramientas analíticas como el Business Explorer (BEx) y herramientas operacionales para importar datos actuales del sistema.

Se pueden considerar varias capas para comprender la estructura de inteligencia empresarial de SAP:

- Capa de Extracción, transformación y carga (ETL): extracción de datos de una fuente determinada, aplicándoles reglas de transformación y cargándolos en el sistema SAP BW.
- Capa de Almacén de datos (Data warehouse area): responsable del almacenamiento de la información en varios tipos de estructuras como por ejemplo, ODS, InfoObjetos y estructuras multidimensionales llamadas InfoCubos.
- Capa de Reporting: esta capa es la responsable del acceso a la información obtenida en la capa de Data Warehouse en forma de informes, los cuales se presentan de una manera sencilla para el analista o el usuario final.
- Capa de planificación (Planning): permite al usuario realizar simulaciones y llevar a cabo tareas como cálculos de presupuesto.

SAP BW es ahora parte de una suite de aplicaciones llamada SAP NetWeaver. Otros componentes de SAP NetWeaver son SAP Enterprise Portal (EP), Web Application Server (WAS), SAP Process Integration (PI) y Master Data Management (MDM). También se incluye herramientas de creación de informes para usuarios finales como el BEx Query Designer y el BEx Analyzer.

La versión 7.0 de SAP BW fue lanzada en junio de 2006 como una parte del SAP NetWeaver 7.0. Este nuevo lanzamiento incluía un gran número de nuevas características, tales como la nueva generación de informes y análisis, mejoras en el almacenamiento de datos y la posibilidad de usar una tecnología innovadora para el desarrollo de las queries de una manera optimizada llamado BI Accelerator (BIA). BIA es una aplicación externa que requiere una licencia adicional.

Recientemente, SAP ha adquirido Business Objects, uno de los primeros desarrolladores de software de informes y aplicaciones. SAP ha indicado que Business Objects operará como una entidad independiente para preservar el principio de agnosticismo de aplicación, pero ha prometido una integración entre SAP BW y Business Objects. Esto supone un nuevo horizonte para la inteligencia empresarial tras la unión de ambas compañías.

Según fuentes de SAP, existen más de 12000 instalaciones de SAP BW.

SAP Business Information Warehouse permite analizar los datos de las aplicaciones SAP operativas, además de otras aplicaciones empresariales y fuentes de datos externas (p. ej., bases de datos, servicios online e Internet). Las funciones de administrador se han diseñado para controlar, supervisar y actualizar todos los procesos de recuperación de datos.

SAP BW activa el Online Analytical Processing (OLAP), que gestiona la información procedente de grandes cantidades de datos históricos y operativos. La tecnología OLAP permite realizar análisis multidimensionales desde varias perspectivas empresariales. El servidor de SAP BW para áreas y procesos básicos, preconfigurado con Business Content, permite ver información de toda la empresa. Para seleccionar roles en una empresa, Business Content ofrece la información que los empleados necesitan para realizar sus tareas. Business Content incluye otros objetos preconfigurados, como InfoCubos, consultas, ratios, características, etc., que facilitan la implementación de SAP BW.

El Business Explorer en SAP BW proporciona herramientas flexibles de análisis y gestión de informes que darán soporte a su empresa a la hora de analizar y tomar decisiones. Entre estas herramientas, se incluyen las funciones de consulta, creación de informes y OLAP. Si es un empleado con permisos de acceso, podrá evaluar datos pasados y actuales en varios niveles de detalle y desde distintas perspectivas, no sólo en la Web sino también en Microsoft Excel.

El diseño de aplicación Web en SAP BW le permite utilizar la navegación OLAP genérica en aplicaciones Web y los Business Intelligence Cockpits para escenarios sencillos y altamente individuales. Puede utilizar los lenguajes de marcado estándar para crear escenarios altamente individuales con elementos de interfase definidos por el usuario. El diseño de aplicaciones Web incluye un amplio espectro de escenarios de BI interactivos basados en Web, que puede ajustar a sus necesidades mediante las tecnologías Web estándar.

4. Planificación temporal y valoración económica

4.1. Planificación temporal

La planificación temporal del proyecto de construcción e implementación del cuadro de mando de Negocios Regulados, está dividida en las siguientes fases:

- Diseño Funcional
- Construcción
- Diseño Técnico
- Pruebas funcionales y de aceptación
- Soporte post-implantación

En la Software Lab de Indra en Lleida, nos hemos ocupado de las fases de construcción, pruebas funcionales y soporte post-implantación, correspondiendo la parte del diseño técnico y funcional a los compañeros analistas de Barcelona.

En la siguiente figura, podemos ver los principales hitos de la planificación del proyecto valorado para tres personas:

Hitos clave	
• Construcción e Implantación del CMNNRR	22-Feb
Diseño Funcional	19-Nov
Construcción	07-Feb
Diseño Técnico	22-Feb
• Pruebas	02-Abr
Pruebas Funcionales	17-Mar
Pruebas de Aceptación	02-Abr
• Soporte Post-Implantación	16-Abr

Figura 3: Hitos en la planificación del proyecto

Como estos plazos son temporalmente amplios y para facilitar una entrega a tiempo, se fijaron varias entregas parciales a lo largo del proyecto. En cada una de estas entregas, se acordó presentar determinadas pantallas, habiendo pasado estas, las pruebas funcionales necesarias para un correcto funcionamiento.

Para finalizar este capítulo, veremos una planificación más visual sobre el tiempo asignado a cada fase.



Figura 4: Esquema temporal de la planificación

Destacar que todos los plazos se cumplieron y actualmente nos encontramos en la fase de mantenimiento y soporte post-implantación del proyecto, para corregir posibles errores o mejoras en la funcionalidad.

4.2. Valoración económica

En este proyecto solo podemos realizar la valoración económica del equipo de trabajo presente en la Software Lab de Indra en Lleida, ya que nuestro trabajo ha sido facturado internamente a Indra, y esta, ya se ha encargado de facilitar al cliente el valor económico del cómputo global de todo el proyecto.

Como comentábamos antes, vamos a mostrar el coste de la faena realizada en este centro por los tres trabajadores dedicados al desarrollo del cuadro de mando.

Un analista programador a jornada completa durante 6 meses facturando a 30 € la hora, considerando 20 jornadas mensuales de 4 horas.

$$30 * 4 * 20 * 6 = 14400$$

Dos programadores junior a jornada completa durante 6 meses con una facturación de 20 € la hora, considerando 20 jornadas mensuales de 8 horas.

$$2 * (20 * 8 * 20 * 6) = 38400$$

$$\text{Coste total} = 14400 + 38400 = 52800 \text{ €}$$

5. Diseño Funcional

5.1. Introducción y objetivos

En este capítulo vamos a abordar el diseño funcional realizado para el desarrollo y la implantación del nuevo Cuadro de Mando de Negocios Regulados (CM NNRR). Este diseño parte de los requerimientos del usuario recopilados y recoge la especificación del sistema en términos de:

- Modelo de Sistema. Descripción del mapa del sistema con los objetos más relevantes. Lista de sistemas que interviene; así como la lista de interfases que los relacionan.
- Modelo de Datos. Entidades que soportan la información relativa al CM NNRR y sus relaciones con el CM Corporativo.
- Interfaz de usuario.
 - Pantallas para la consulta de la información del CM NNRR (Web-Reporting)
 - Parrillas de entrada de los datos manuales del CM NNRR (SEM-BPS)
 - Programas de administración del CM NNRR

Los objetivos del quinto capítulo son:

- Definir la estructura de la información del sistema para la creación del modelo de datos.
- Definir las pantallas y procesos asociados para el manejo de la información por parte del usuario.
- Establecer los flujos de información entre componentes del sistema.

En este capítulo no nos dedicaremos a crear el análisis funcional y el diseño técnico, puesto que estos dos documentos corresponden a los analistas que tienen trato con el cliente. En las Software Labs de Indra, recibimos el diseño hecho y nos dedicamos a realizar la implementación basándonos en dicho diseño técnico.

5.2. Modelo de sistema

El CM NNRR se desarrollará en el sistema actual de Control de Gestión SAP-SEM-BW; donde residen también las aplicaciones de Reporting Operativo y de Actividades (Presupuesto y Seguimiento) y los Cuadros de Mando Corporativos y de Negocio.

Extraemos información de diez sistemas origen diferentes, los cuales no nombraremos por temas de confidencialidad.

Todos estos sistemas Origen, menos obviamente el sistema SAP-SEM-BW, son sistemas externos. En la medida de lo posible se cargará la información de estos sistemas externos al Sistema DWH y posteriormente se extraerá desde el sistema SAP-SEM-BW la información necesaria. Esta información necesaria estará ubicada en el Repositorio de Datos (Tablas de la BD Oracle) del Sistema DWH; repositorio creado especialmente para las extracciones del CM NNRR. Los indicadores de otro sistema origen (SO_1) se extraerán mediante herramientas propias de BW al ser el sistema SO_1 un sistema SAP-BW.

El resto de extracciones, que no se carguen en el DWH, se realizarán vía ficheros. Los ficheros se extraerán del servidor de ficheros de cargas que habrán sido anteriormente transferidos desde los sistemas externos vía FTP.

Para el sistema origen SAP-BW-SEM se crearán los procesos necesarios para cargar la información Económico-Financiera necesaria para la formulación de los indicadores. Dicha información corresponderá al ámbito del negocio Negocios Regulados.

El siguiente grafico muestra el modelo de sistemas con los sistemas que intervienen y sus relaciones:

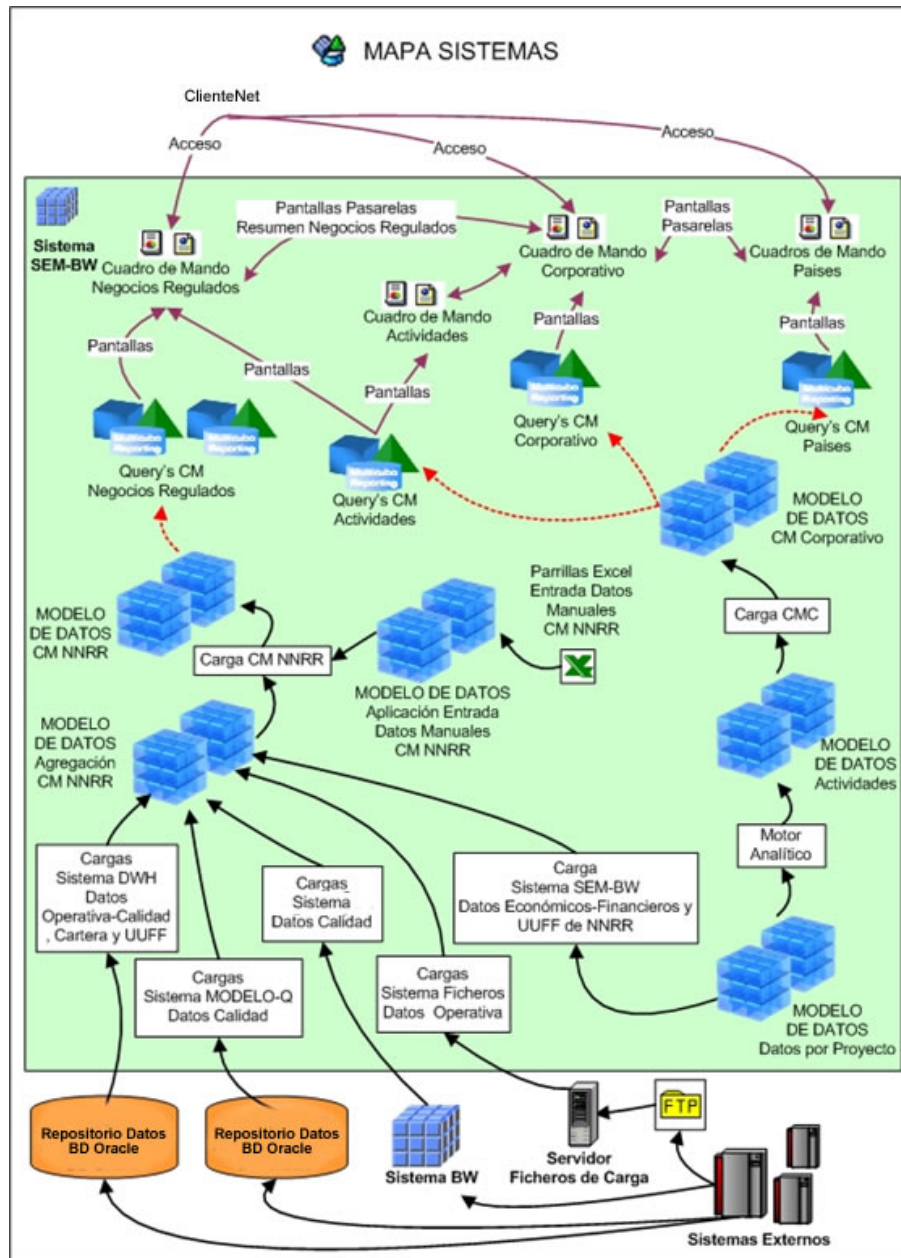


Figura 5: Mapa de Sistemas

5.3. Modelo de datos

En esta sección describiremos con detalle, las funcionalidades requeridas, de forma que el diseñador técnico pueda proceder al diseño de su construcción. Para empezar mostraremos un diagrama del modelo de datos sobre el que vamos a trabajar.

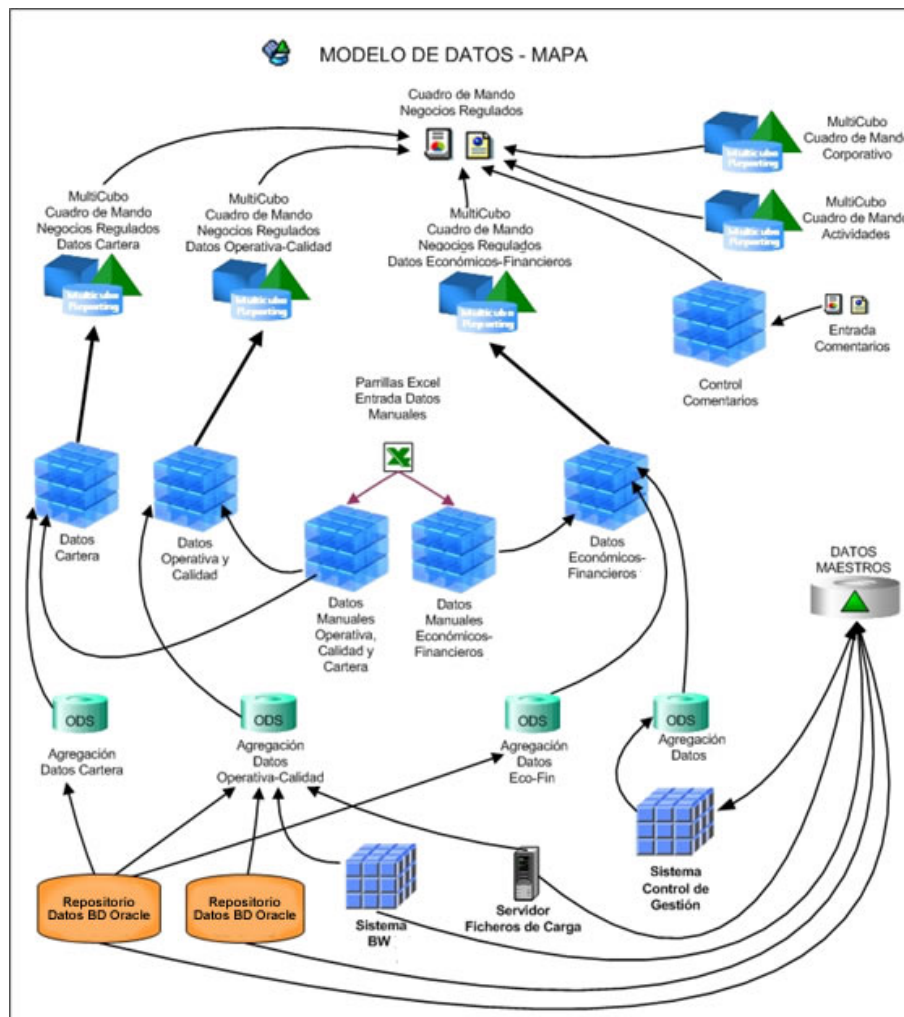


Figura 6: Modelo de datos

En el diagrama anterior, los datos maestros estarían distribuidos de la siguiente forma:

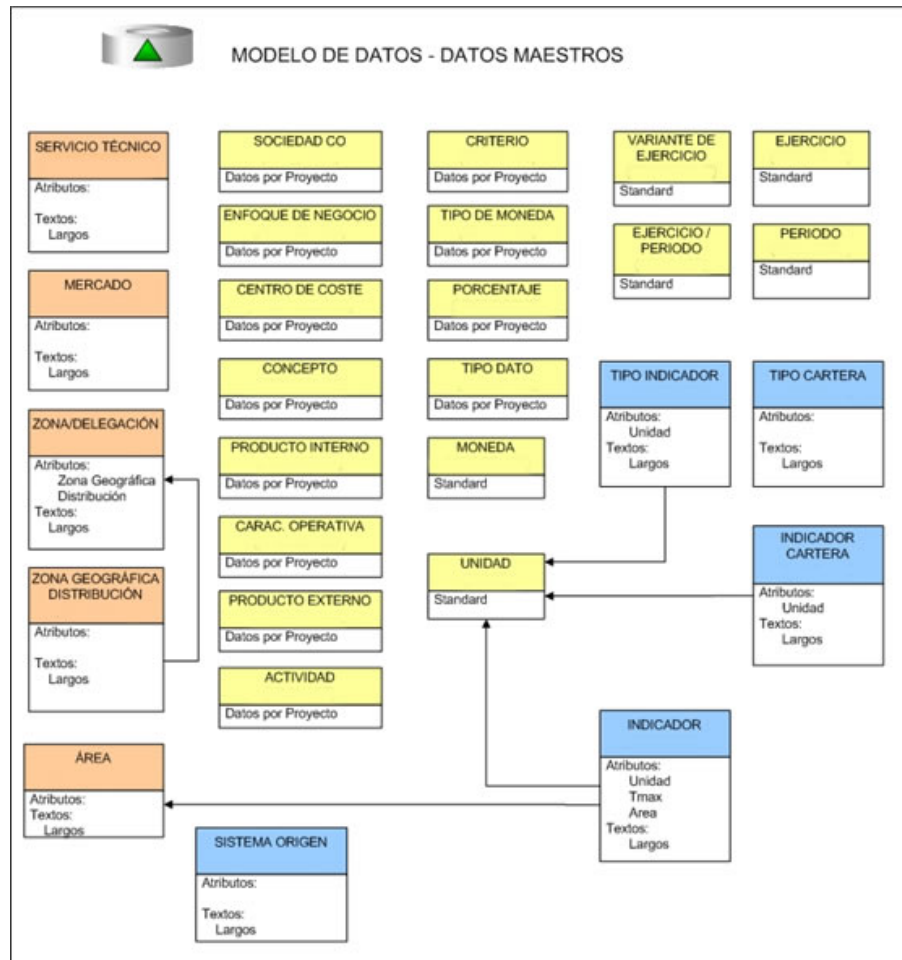


Figura 7: Datos maestros del modelo de datos

A continuación describiremos los InfoObjetos más significativos:

- Área: Es el área de operaciones.
- Zona/Delegación: Zona o Delegación donde se realiza la distribución/expansión del negocio
- Servicio Técnico: Servicio Técnico que mantiene las Operaciones
- Mercado: mercados en los que opera el negocio
- Sistema Origen: Define el origen de la información
- Indicador: indicador para evaluar el negocio

- Tipo Indicador: tipo de información que evalúa el indicador
- Resto de Características(Sociedad CO, Centro de Coste,...) marcadas en el gráfico en amarillo, son las mismas definidas en el Reporting Operativo (Presupuesto y Realidad) del sistema de Control de Gestión SEM-BW.

5.4. Interfaz de usuario

En este apartado trataremos el diseño inicial de las pantallas y en que documentos nos hemos basado para obtener las pantallas finales con las que trabajará el usuario.

El formato de las pantallas de la aplicación queda descrito en una maqueta en PowerPoint que nos facilitó el cliente y sobre la cual nos basaremos en la creación de esta parte del proyecto.

La descripción de las pantallas queda descrita en los archivos Excel de formulación, los cuales, nos indicarán que InfoObjetos tenemos que usar para crear cada Query que posteriormente mostraremos en la pantalla.

Empezaremos comentando como utilizar la información que tenemos en la formulación para diseñar las queries. Para ello mostraremos un archivo de Excel con la correspondiente información de la query e indicaremos como usarla en el Query Designer¹

Microsoft Excel - IndraDF_01.03.Form_Italia_v.8.0(definitiva).xls

Archivo Edición Ver Insertar Formato Herramientas Datos Ventana ?

Escriba una pregunta

H0

1 Pantalla 1 - SubPantalla 1

2

3 MultiCubo ICNRM_CM

4

5

6

7

8

9 Tabla 1

10 MultiCubo

11 Descrip

12 Fila

13 Columna

14

15

16 En Miles, P,EU y Criterio Corporativo

17 Fila 1

18 SubFila 11

19 SubFila 12

20 Fila 2

21 Fila 3

22 Fila 4

23 Fila 5

24 Fila 6

25 Fila 7

26 Fila 8

27 Fila 9

28 Fila 10

29 Fila 11

30 Fila 12

31 Tabla 2

32 MultiCubo

33 Descrip

34 Fila

35 Columna

36

37 En Miles, P,EU y Criterio Corporativo

38 Fila 1

39 Fila 2

40 Fila 3

41 Fila 4

42 Fila 5

Figura 8: Documento de formulación de las queries

Como podemos ver en la imagen, la formulación nos indica los datos desde el nombre de la pantalla hasta los filtros que usará la query, pasando por el valor de cada característica o ratio utilizados en la consulta.

¹Programa en el que profundizaremos en el capítulo siguiente, 6.2.2

En la columna descripción, tenemos el nombre de la tabla y el de cada fila de la misma. En esta columna, con fondo naranja, tenemos el factor de graduación de la tabla, es decir, si muestra valores en unidades, miles, etc. La columna unidad, nos muestra el ratio, indicando si estamos ante una cantidad o un importe. Dentro de formulación, en Estructura Organizativa, contiene los filtros que usaremos en las características libres de la query. El concepto muestra el valor que tendrá la query y la jerarquía en la que tendremos que buscar dicho valor. En el caso que este valor se obtenga por medio de una fórmula, esta se indicará en esta misma columna. Por último, citar que cada pestaña de la hoja de cálculo se corresponde con cada una de las pantallas de este grupo de pantallas.

La maquetación de las pantallas ha sido diseñada por el cliente en PowerPoint, nos muestra principalmente la situación de las tablas y del gráfico (solo habrá uno por pantalla). Asimismo, obtendremos de este archivo, el contenido de los selectores de gráfico o tabla, los colores a usar en el gráfico y la cabecera con el nombre de la pantalla y de los selectores de cabecera que usaremos para filtrar.

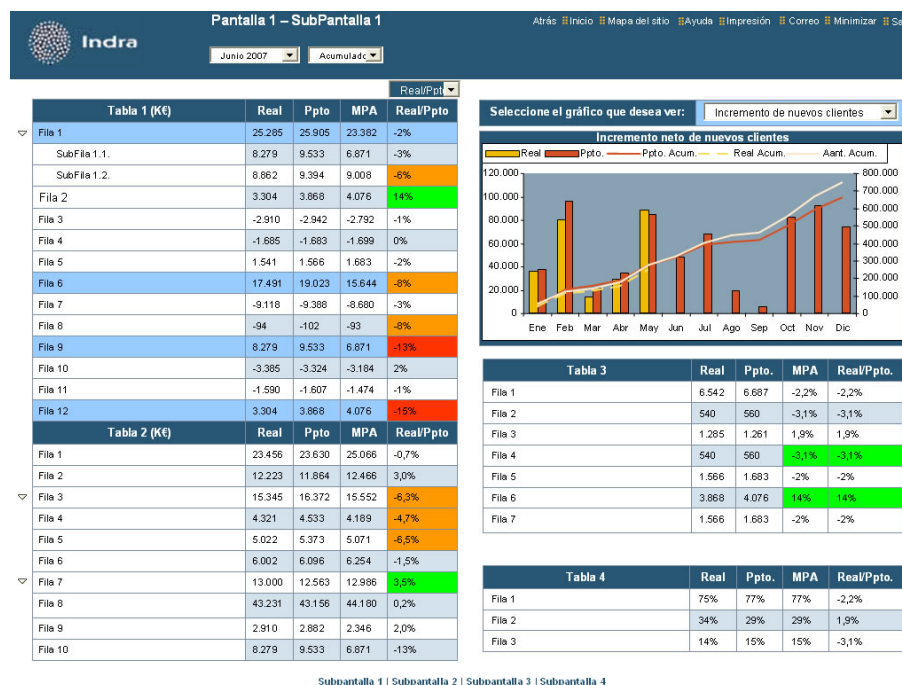


Figura 9: Documento de maquetación de las pantallas

6. Tecnologías utilizadas

6.1. Lenguajes de programación usados

Para crear el cuadro de mando de negocios regulados pretendido por el cliente, hemos hecho uso de los siguientes lenguajes de programación: ABAP, HTML y JavaScript.

A continuación, pasaremos a hablar brevemente de los lenguajes en sí, y de que uso hacemos de ellos en nuestra aplicación.

6.1.1. ABAP

ABAP (Advanced Business Application Programming) es un lenguaje de cuarta generación, propiedad de SAP, que se utiliza para programar la mayoría de sus productos (R/3, mySAP Business suite...). Utiliza sentencias de Open SQL para conectarse con prácticamente cualquier base de datos. Cuenta con miles de funciones para el manejo de archivos, bases de datos, fechas, etc... Permite conexiones RFC (Remote Function Calls) para conectar a los sistemas SAP con cualquier otro sistema o lenguaje de programación.

ABAP fue desarrollado por SAP como lenguaje de informes para SAP R/2 en los años 80, una plataforma que permitía a las grandes corporaciones construir aplicaciones de negocios para gestión de materiales y finanzas.

ABAP, muy parecido al COBOL en sus orígenes, originalmente significaba Allgemeiner Berichtsaufbereitungsprozessor, palabras alemanas para procesador genérico para la preparación de informes. En sus inicios incluía el concepto de Bases de datos lógicas, que suministraba un alto nivel de abstracción para el acceso a bases de datos.

ABAP fue pensado como un lenguaje de programación para que los usuarios finales pudieran manipular la información, pero el 4GL se fue volviendo demasiado complicado para usuarios normales, por lo que es necesario programadores experimentados para realizar desarrollos.

ABAP se mantuvo como el lenguaje de desarrollo para la siguiente versión cliente-servidor de SAP R/3, que fue lanzada al mercado en 1992, en el que casi todo el sistema, menos las llamadas al sistema básicas estaban escritas en ABAP. En 1999, con el lanzamiento de la versión 4.5 de R/3, SAP lanzó una extensión orientada a objetos denominada ABAP Objects.

La última plataforma de desarrollo de SAP, NetWeaver, soporta ABAP y Java como lenguajes de programación.

Donde y como lo usaremos en el proyecto

Este lenguaje lo hemos utilizado para crear programas que usaremos para desarrollar las pantallas o cargar datos para mostrarlos posteriormente en las pan-

tallas. Es decir, la parte visible del proyecto, no está creada con ABAP, pero sí, la rutina para cargar datos con las cadenas, obtener datos de un servidor externo o modificar la estructura de las pantallas finales desde la transacción SE38.

6.1.2. HTML

HTML, siglas de HyperText Markup Language (Lenguaje de Etiquetas de Hipertexto), es el lenguaje de marcado predominante para la construcción de páginas web. Es usado para describir la estructura y el contenido en forma de texto, así como para complementar el texto con objetos tales como imágenes. HTML se escribe en forma de "etiquetas", rodeadas por corchetes angulares (<,>). HTML también puede describir, hasta un cierto punto, la apariencia de un documento, y puede incluir un script (por ejemplo Javascript), el cual puede afectar el comportamiento de navegadores web y otros procesadores de HTML.

Por convención, archivos de formato HTML usan la extensión .htm o .html.

Donde y como lo usaremos en el proyecto

El diseño de las pantallas finales que utilizarán los usuarios se ha realizado en HTML. En estas pantallas, mostramos la información obtenida de las queries y les damos el formato necesario en HTML. Los datos obtenidos serán mostrados en tablas y gráficos, los cuales los posicionaremos en la pantalla mediante el lenguaje de etiquetas. También lo usaremos para mostrar y utilizar los selectores gracias a los cuales elegiremos que gráfico o campos de la tabla mostrar.

6.1.3. JavaScript

JavaScript es un lenguaje de programación interpretado, es decir, que no requiere compilación, utilizado principalmente en páginas web, con una sintaxis semejante a la del lenguaje Java y el lenguaje C.

Al igual que Java, JavaScript es un lenguaje orientado a objetos propiamente dicho, ya que dispone de Herencia, si bien esta se realiza siguiendo el paradigma de programación basada en prototipos, ya que las nuevas clases se generan clonando las clases base (prototipos) y extendiendo su funcionalidad.

Todos los navegadores modernos interpretan el código JavaScript integrado dentro de las páginas web.

El lenguaje fue inventado por Brendan Eich en la empresa Netscape Communications, que es la que desarrolló los primeros navegadores web comerciales. Apareció por primera vez en el producto de Netscape llamado Netscape Navigator 2.0.

Tradicionalmente, se venía utilizando en páginas web HTML, para realizar tareas y operaciones en el marco de la aplicación únicamente cliente, sin acceso a funciones del servidor. JavaScript se ejecuta en el agente de usuario al mismo tiempo que las sentencias van descargándose junto con el código HTML.

Donde y como lo usaremos en el proyecto

El lenguaje javascript lo usaremos en el diseño de pantallas para completar al código HTML y realizar efectos que no podemos realizar con el HTML, como pueden ser dar formato a tablas y gráficos, ocultar elementos de la pantalla para mostrar otros en su lugar o inicializar la pantalla para que cada vez que se cargue la tengamos mostrando siempre los mismos datos y con el mismo formato.

En uno de los apéndices del proyecto, adjuntaremos las funciones javascript utilizadas en el desarrollo de las pantallas.

6.2. Programas utilizados

Para desarrollar este proyecto, hemos usado cuatro programas, a continuación, comentaremos el uso de cada uno a lo largo del proyecto y como utilizarlos mediante ejemplos útiles para los diferentes casos ante los que tendremos que hacer frente.

Como la confidencialidad es una tema importante cuando se trabaja con el cliente, no utilizaremos para este documento, todo el código realizado, sino que expondremos la forma de utilizar estos programas y como hemos comentado antes, varios ejemplos. Podríamos decir, que de cada programa, mostraremos un ¿Qué hace? Y un ¿Cómo se hace?

6.2.1. Sap Logon

Mediante el Sap Logon, nos conectamos a la interfaz gráfica de SAP (SAP GUI) en el servidor de nuestro cliente, servidor sobre el cual trabajaremos también con los otros programas.

Una vez conectados en el servidor de nuestro cliente, trabajaremos con las diferentes transacciones de SAP, de las cuales vamos a explicar las que hemos utilizados en este proyecto y que por lo general son las más usadas en el campo de la Inteligencia Empresarial.

6.2.1.1. RSA1: Administrator Workbench

La transacción de SAP BW, Administrator Workbench (RSA1) abreviada como AWB, es la herramienta principal para tareas en el proceso de almacenamiento de datos. Esto proporciona funciones de modelado de datos, así como funciones para el control, supervisado y mantenimiento de todos los procesos de SAP BW que tengan que ver con la consecución, retención y proceso de datos.

Estas funciones son ocho, pero solo nos expandiremos en la función de modelado que es la que hemos utilizado en el desarrollo del proyecto.

6.2.1.1.1. Modelado: En modelado, podemos crear y editar todos los objetos y reglas del administrador de Workbench que necesitamos para el transporte, actualización y análisis de datos.

Los objetos se muestran en una estructura de árbol ordenados por jerarquías. Usando el menú contextual de objetos, podemos seleccionar el menú de mantenimiento correspondiente a cada uno o realizar las funciones más relevantes para ese tipo de objeto.

InfoProvider: Un InfoProvider es un objeto para el cual las queries pueden ser creadas o ejecutadas en BEX. Asimismo son los objetos o las vistas que son relevantes para la realización de informes.

Los InfoProviders son meta objetos de la base de datos que uniformemente puede ser vista como proveedores de datos dentro de una definición de query², y cuyos datos también pueden ser relatados uniformemente. El tipo de recuperación de datos, el grado de detalle "o proximidad" al sistema fuente en el diagrama de flujo de datos se diferencia de un InfoProvider a otro. Estas diferencias no afectan al report. En el Reportaje, estos son objetos que contienen los datos que son convenientes para la evaluación que usa una query de BEX.

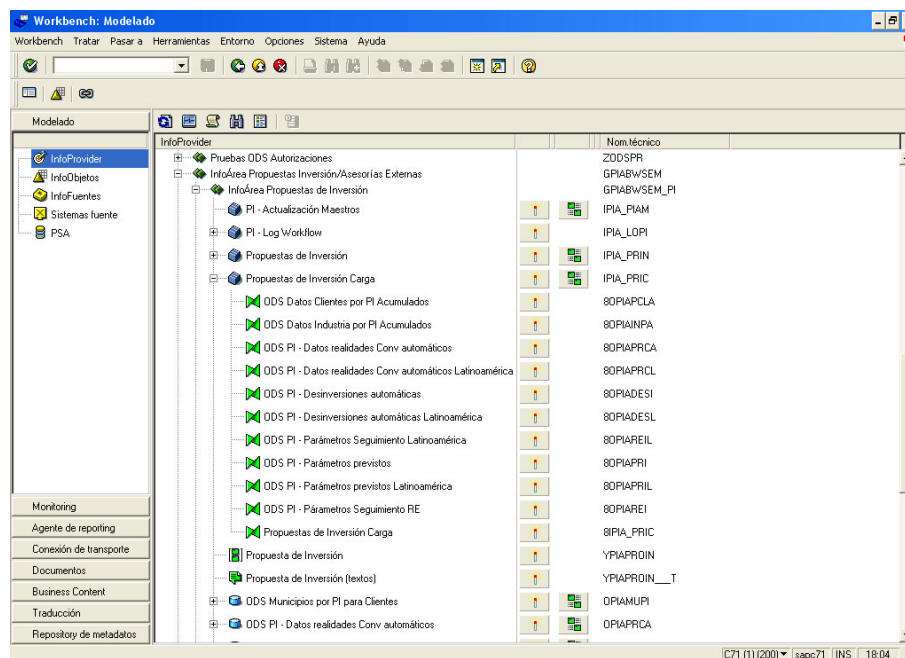


Figura 10: RSA1: Menú de InfoProvider

En la pestaña de InfoProviders, RSA1, contiene los objetos que físicamente abarcan datos:

- **Infocubo básico:** Estructura que almacena datos que posteriormente usaremos en BW. Como el resto de InfoCubos, la estructura del InfoCubo Básico corresponde al Esquema de Estrella. Se llena mediante las reglas de actualización definidas

²Consulta. Inserción, actualización, búsqueda o eliminación en una base de datos.

- **Infocubo transaccional:** se diferencia de los InfoCubos Básicos en su capacidad de apoyar el acceso paralelo a la escritura. Los InfoCubos Básicos técnicamente están optimizados para accesos leídos en detrimento de los accesos escritos.
- **Objetos ODS:** DataStore Objects almacenan datos limpios y consolidados en un nivel detallado. En reporting se utilizan ODS para realizar reportes detallados en SAP BI. Un ODS describe un set de datos consolidados desde una o más fuentes de datos. Los ODS están disponibles para datos transaccionales y para datos maestros. Se puede analizar este set de datos en el BEx Query. En contraste con el almacén de datos para infocubos multidimensionales, los datos para ODS se almacenan en tablas planas. Un ODS contiene una clave, tanto como campos de datos, que pueden contener ratios como características. Se puede transferir datos del ODS en los InfoCubos conectados o en un ODS adicional en el mismo sistema o en un sistema diferente.

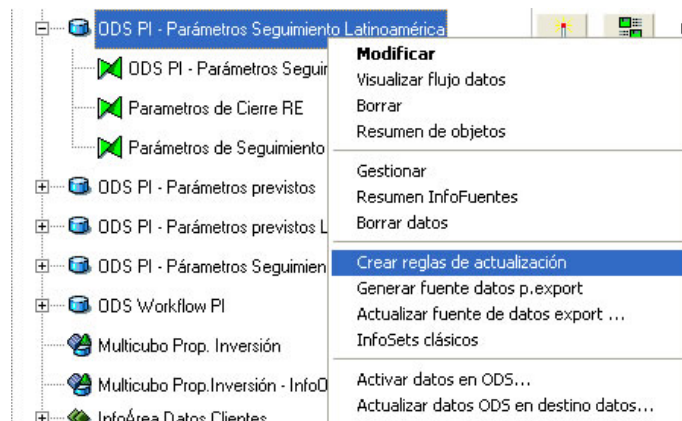


Figura 11: RSA1: Crear reglas de actualización de un ODS

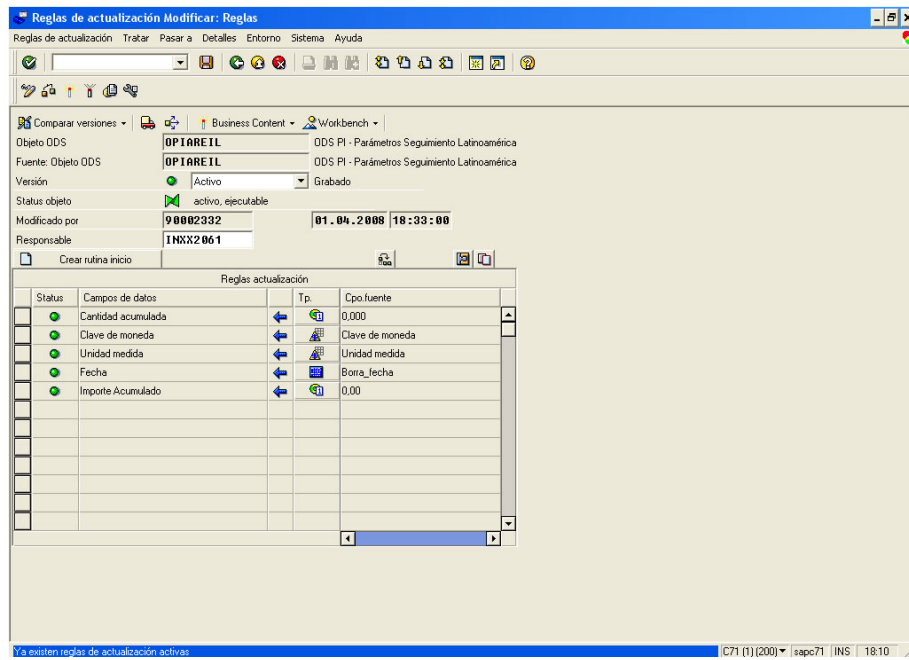


Figura 12: RSA1: Crear reglas de actualización de un ODS - 2

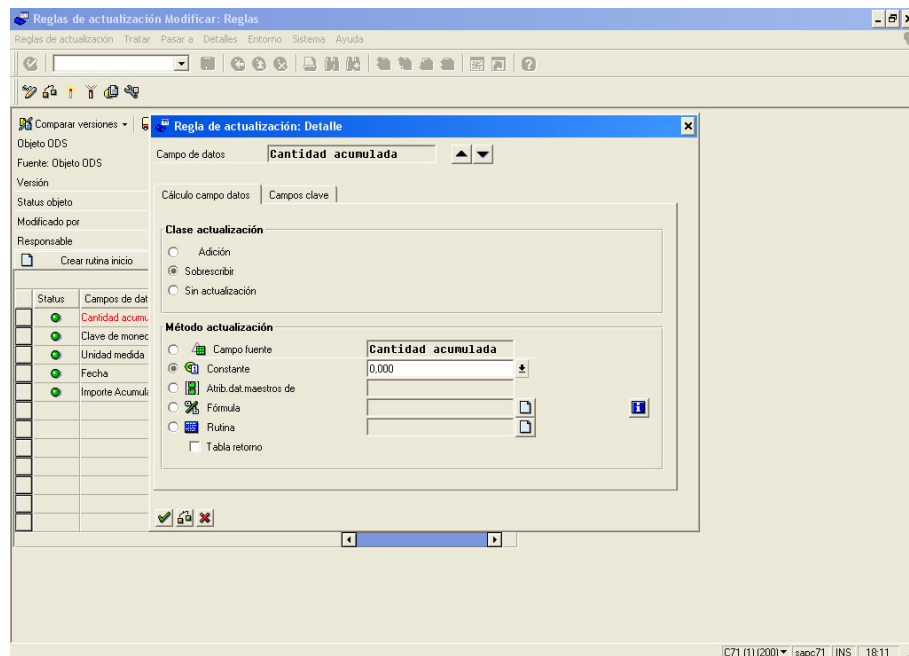


Figura 13: RSA1: Crear reglas de actualización de un ODS - 3

- **InfoObjetos:** Un InfoObjeto característica puede tener tablas en las cuales la información de los datos maestros (atributos, textos, jerarquías) es cargada desde el sistema fuente. Una vez que se haya incorporado un InfoObjeto característica en el árbol de InfoSitios, es posible utilizarla como un InfoSitio para el BEx Reporting.
- **InfoSets:** Los BI InfoSets no poseen un almacén de datos para ellos mismos, pero en cambio conectan tablas planas utilizando uniones, entonces proveen una capa semántica adicional (vista relevante del reporting) para el BEx reporting. Se pueden crear BI InfoSets desde InfoCubos (sólo del tipo Standard), DataStore Objects e InfoObjetos características con datos maestros.
- **Cubos Remotos:** es un InfoCubo cuyos datos de transacción no son manejados en la inteligencia empresarial. Sólo la estructura del cubo remoto es definida en BW. Los datos son leídos para relatar la utilización de un BAPI de otro sistema.
- **InfoCubos Virtuales:** es un InfoCubo que físicamente no almacena sus propios datos en BW. La fuente de datos es un módulo de función definido por usuario. Tenemos un número de opciones para definir las propiedades

de la fuente de datos con más precisión. Depende de estas propiedades, el manejo de datos proporciona servicios para convertir los parámetros y datos.

- **MultiProviders:** Un Multiprovider combina datos desde diferentes InfoSitios y los hace accesibles para ser utilizados en los reportes. El MultiProvider no contiene datos en sí mismo. Los datos provienen exclusivamente de los InfoSitios en los cuales se basa. Se puede crear un MultiProvider desde cualquier combinación de InfoSitios.

InfoObjetos: Los objetos de evaluación de negocio en BW son conocidos como InfoObjetos. Se dividen en características (por ejemplo, clientes), figuras clave (por ejemplo, el ingreso), unidades (por ejemplo, la moneda, la unidad de cantidad), características de tiempo (por ejemplo, el año fiscal) y características técnicas (por ejemplo, el número de solicitud). Son las unidades más pequeñas de la información en BW. Estructuran la información necesaria para crear destinos de datos.

InfoFuentes: En BW, una InfoFuente describe la cantidad de todos los datos disponibles para una transacción de negocio o un tipo de transacción de negocio (por ejemplo, la contabilidad de centro de coste).

Una InfoFuente es una cantidad de información que lógicamente se representa como una. Esto prepara datos consolidados para poner al día a los destinos de datos. Contienen datos de transacción o datos maestros (atributos, textos y jerarquías).

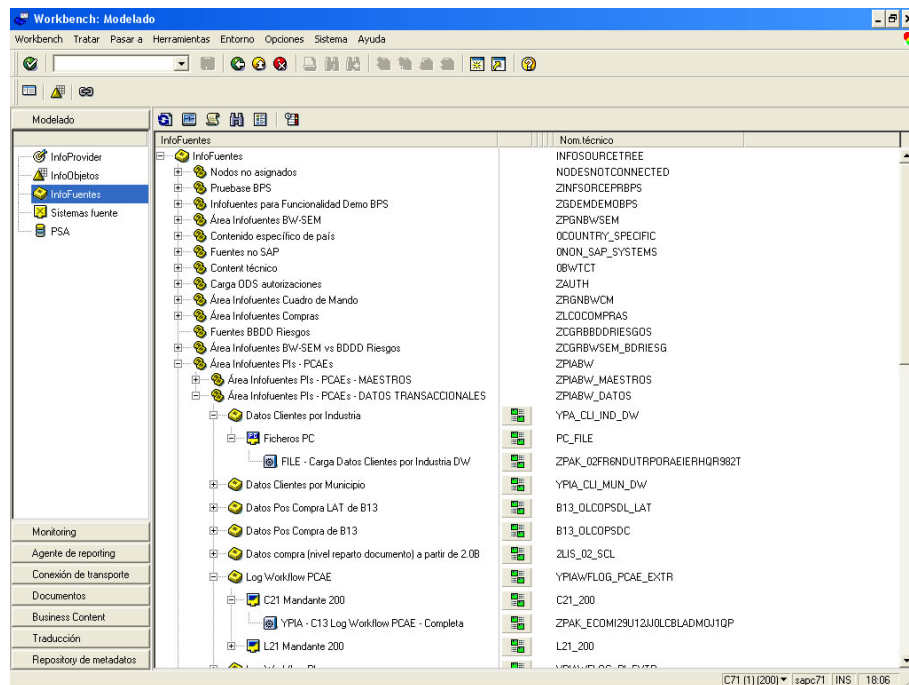


Figura 14: RSA1: Menú de InfoFuentes

Una vez hemos creado la InfoFuente, le asignamos una fuente de datos, que será una de las que veremos en la pestaña de sistemas fuente que explicaremos en el siguiente punto.

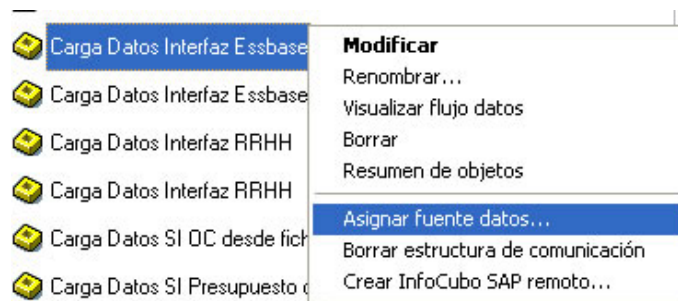


Figura 15: RSA1: Asignar fuente de datos a una InfoFuente

Una vez tenemos asignado un sistema fuente a la InfoFuente, crearemos un InfoPaquete, que lo utilizaremos para la carga de datos. Este InfoPaquete contendrá la información necesaria sobre su origen y destino de datos y el proceso necesario para llevar los datos al ODS o InfoCubo.

En la InfoFuente, hemos de crear unas reglas de transferencia, las cuales usamos para determinar como queremos que los campos transferidos de la estructura origen sean asignados al InfoObjeto.

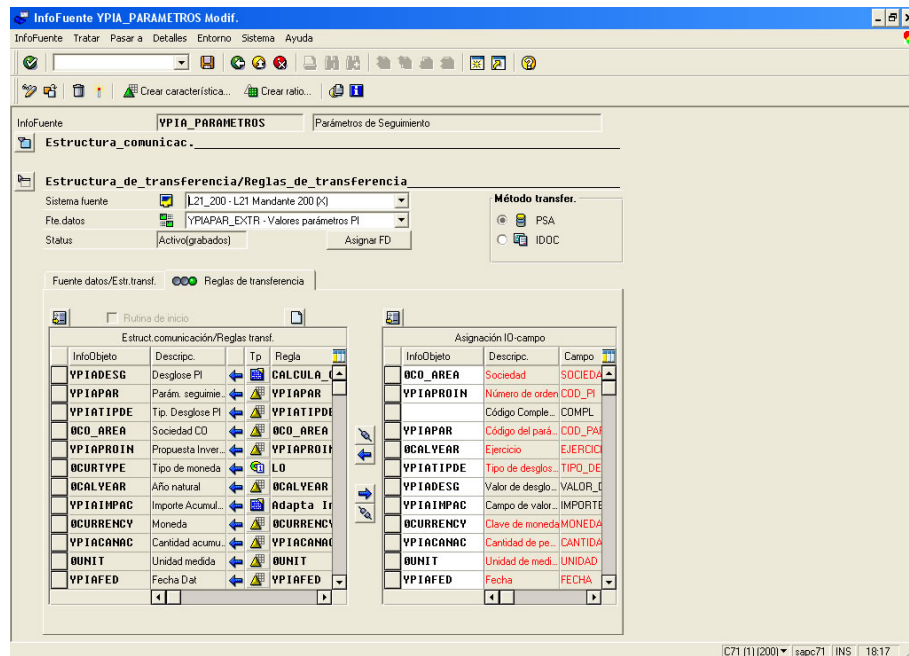


Figura 16: RSA1: Reglas de transferencia de una InfoFuente

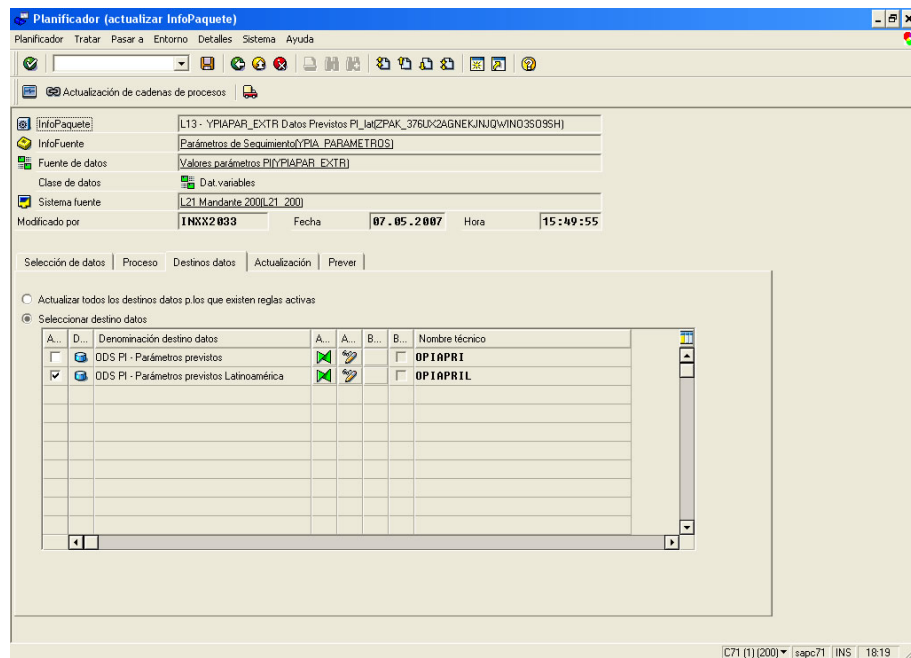


Figura 17: RSA1: Infopaqete asignado a una InfoFuente

Sistemas fuente: Son los sistemas de los que extraemos la información que posteriormente usaremos en las InfoFuentes. Estos sistemas pueden ser entre otros un sistema de ficheros del PC, un sistema de bases de datos o un sistema de un servidor externo.

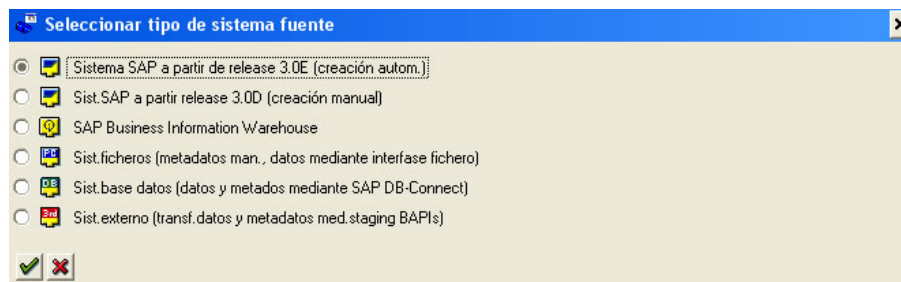


Figura 18: RSA1: Tipos de Sistemas Fuente

PSA: El Organizar Persistente de Área (PSA) es el área de almacenamiento entrante para datos de los sistemas fuente de SAP BW. Los datos solicitados son guardados sin cambios respecto a los del sistema fuente.

Los datos requeridos son almacenados en el formato de estructura de transferencia en tablas de bases de datos relacionales en BI. El formato de datos permanece inalterado, queriendo decir que ningún resumen o transformación ha ocurrido.

Si seleccionamos PSA cuando extraemos datos, conseguimos el funcionamiento mejorado si usamos TRFCS para cargar los datos. La facilidad de almacenaje temporal en el PSA también nos permite comprobar y cambiar los datos antes de la actualización en objetivos de datos. En contraste con una petición de datos con IDOCS, una petición de datos en el PSA también nos da varias opciones para una actualización remota de los datos en el destino de datos.

6.2.1.2. SE11: ABAP Dictionary

Desde SE11, podemos crear y modificar tablas, vistas, tipos de datos o dominios entre otros.

Es útil para visualizar datos de tablas para ver que tenemos cargado en las mismas y comprobar si las queries funcionan correctamente.

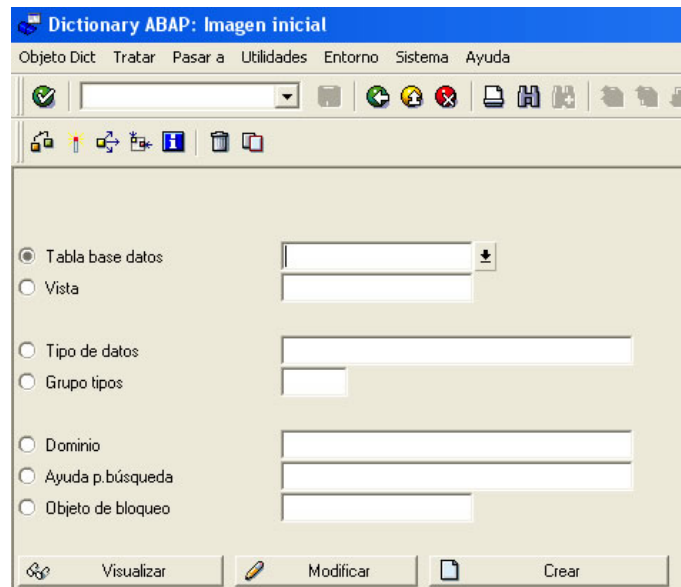


Figura 19: SE11: Diccionario ABAP

6.2.1.3. SE38: Program Editor

Esta transacción se utiliza entre otros usos para modificar los programas que hemos creado en ABAP. Nosotros la hemos utilizado principalmente para modificar las pantallas finales que tendrá la aplicación con el programa que habíamos creado como editor de modelos de pantalla.

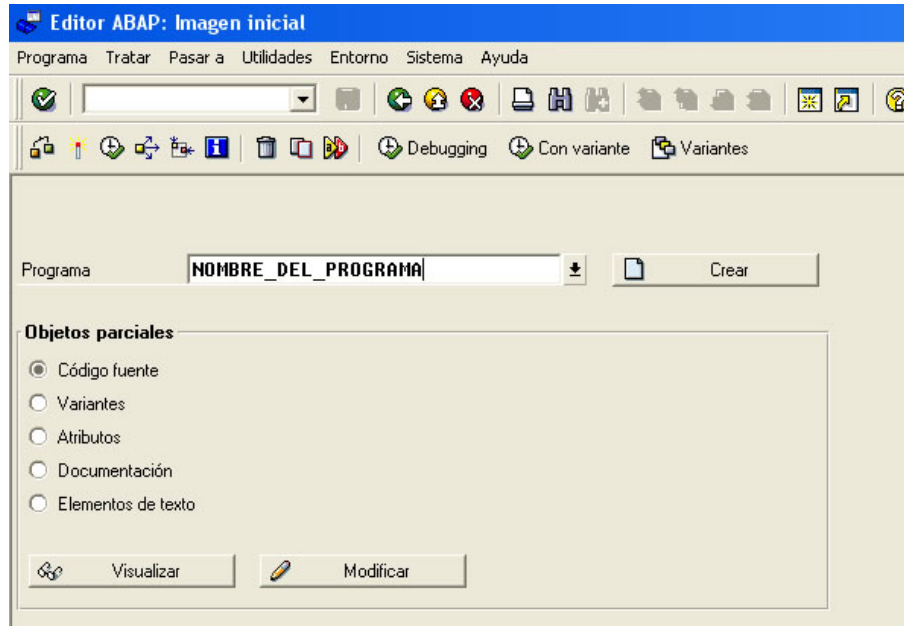


Figura 20: SE38: Modificar programa

Una vez, hemos seleccionado nuestro programa, si pulsamos F8 (ejecutar) , vamos al editor de modelos, una vez allí, introducimos en el cuadro de texto, la pantalla que queremos modificar y pulsamos de nuevo F8.



Figura 21: SE38: Modificar pantalla

Una vez aquí, podemos modificar la pantalla creada anteriormente o hacer una nueva a partir de esta.

El principal inconveniente de utilizar esta transacción para editar programas o pantallas, es que no diferencia con colores, las palabras reservadas del lenguaje de programación correspondiente. Solo diferencia los comentarios de los programas creados en ABAP.

6.2.1.4. SE80: Object Navigator

SAP's ABAP Workbench (SE80) es un conjunto de herramientas y librerías para diseñar, implementar, probar y gestionar transacciones y reports escritos con objetos de ABAP.

En SE80 tenemos los siguientes buscadores:

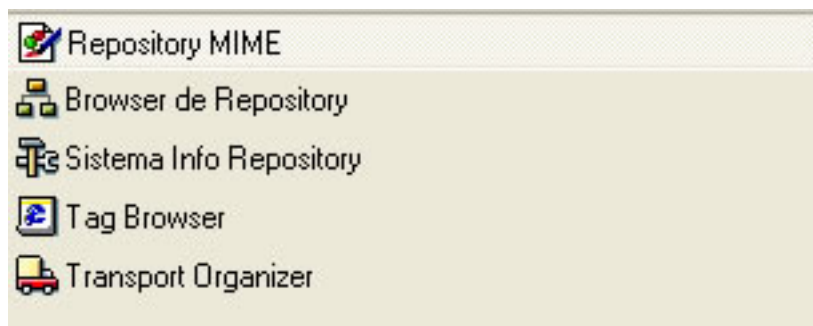


Figura 22: SE80: Browser

6.2.1.4.1. Repository MIME Muestra todos los directorios con objetos MIME que serán importados al sistema actual.

6.2.1.4.2. Browser de Repository Es el browser que aparece de inicio al abrir la transacción. SE80.

Nos muestra una colección de objetos en forma de lista de objetos. Estos objetos se seleccionan por categoría (paquetes, programas, clases, objetos locales, etc). Con permisos de desarrollador, podemos modificar los que no sean standards de SAP.

6.2.1.4.3. Sistema Info Repository Muestra todos los objetos del sistema de información sin la preselección.

6.2.1.4.4. Tag Browser Muestra etiquetas para aplicaciones Web. Podemos limitar la lista con la etiqueta relevante en aplicaciones ITS o en aplicaciones BSP con preselección.

6.2.1.4.5. Transport Organizer Solicita las salidas en el sistema actual.

6.2.1.5. RSPC: Process Chain and Process Variant

Transacción utilizada para crear, modificar y administrar las cadenas de carga de datos. Como su propio nombre indica, es el proceso o rutina que siguen los datos desde el sistema origen hasta llegar a su destino (ODS infoobjeto o infocubo) y desde los cuales serán utilizados en las queries para posteriormente ser mostrados por pantalla.

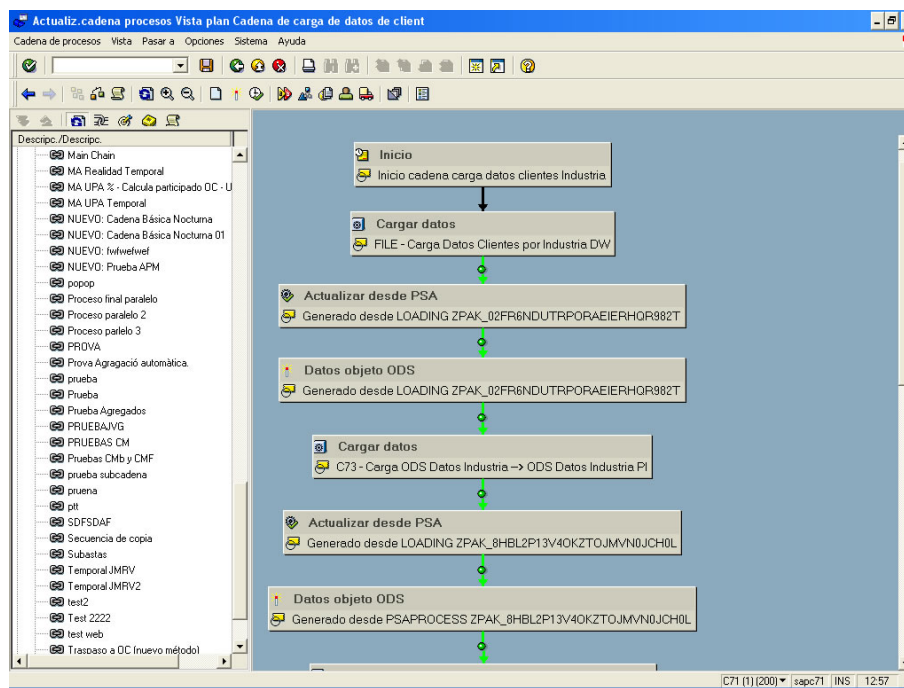


Figura 23: RSPC: Ejemplo de cadena de carga

6.2.1.6. RSRT: Monitor query Support Package

Una query basada en un MultiProvider se divide internamente en sub-queries. Una sub-query es generada a partir de cada InfoProvider asociado con el Multi-Provider.

La división de una query MultiProvider en sub-queries puede ser bastante compleja. Si hemos definido una query para un MultiProvider y queremos mostrar como ha sido subdividida, necesitamos llamar a la transacción RSRT. Esto puede ser un paso útil si la query no se comporta como esperábamos.

6.2.1.7. SMICM: ICM Monitor

Esta transacción se utiliza para monitorizar y administrar la comunicación con Internet, y los datos que envía y recibe de este.

El principal uso que hemos dado de esta transacción ha sido para borrar la caché del servidor de Internet cada vez que hacíamos un cambio en las funciones javascript utilizadas en las pantallas.

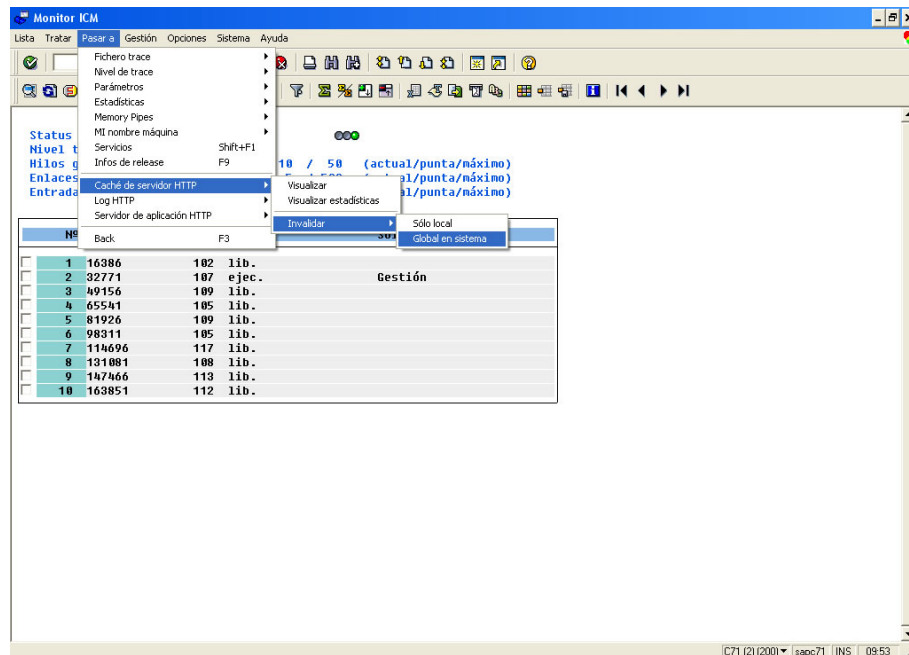


Figura 24: SMICM: Invalidar Caché del servidor

6.2.1.8. SM30: Maintain Table Views

Transacción de uso similar a SE11, pero esta está limitada al mantenimiento y visualización de tablas.

6.2.2. Query Designer

El programa BEx Query Designer es el que utilizaremos para crear las queries que posteriormente utilizaremos para mostrar los datos en forma de tabla o gráfico en las pantallas a las que tendrán acceso los usuarios finales de nuestro cliente.

Estas queries las crearemos a partir de los infoproviders que hemos creado previamente desde la transacción RSA1. En las queries seleccionaremos y combinaremos Infoobjetos (características, características de tiempo, ratios...) o reutilizaremos elementos que hayan sido creados previamente en otras.

Se parametrizarán las queries utilizando las variables de las características, jerarquías, textos, fórmulas... Las variables son parámetros de una query que se definen en el Query Designer y que se rellenarán con valores al ejecutarla.


Los componentes más significativos dentro de una query son los filtros y el poder navegacional que presentan los objetos introducidos.

Esta herramienta es un componente de SAP BW, que permite crear queries, combinando características y ratios definidos en el modelo OLAP (InfoProvider).

6.2.2.1. Áreas de trabajo en el Query Designer Además de la barra de Herramientas el Query Designer está compuesto por las siguientes áreas de trabajo:

- Infositio: nos muestra las estructuras, ratios y características del multicubo correspondiente, las cuales podemos utilizar en la creación de la consulta.
- Columnas: son las columnas que aparecerán en la tabla. En el caso de una query de gráfico, en la sección de columnas, tendremos el intervalo de tiempo que mostrará dicho gráfico.
- Líneas: filas que mostrará la tabla. Si estamos ante una query de gráfico, a parte de las filas, tendremos, las columnas de la misma query de gráfico.
- Características libres: características comunes para todas las posibilidades de combinación entre filas y columnas, a no ser que haya definición celda a celda.
- Filtros: características por las que podremos filtrar la query definitiva y obtener una consulta diferente.
- Presentación preliminar: esbozo de cómo quedará la tabla con este diseño.

6.2.2.2. Crear una query nueva

Para crear una nueva query, se debe presionar el botón , que abrirá una ventana de dialogo desde donde se debe seleccionar el Infoprovider que definirá las características, Ratios y características de tiempo que van a definir a la query. Además de eso vamos a poder definir Ratios calculados y restringidos, a partir de los ratios que componen al cubo, así como fórmulas.

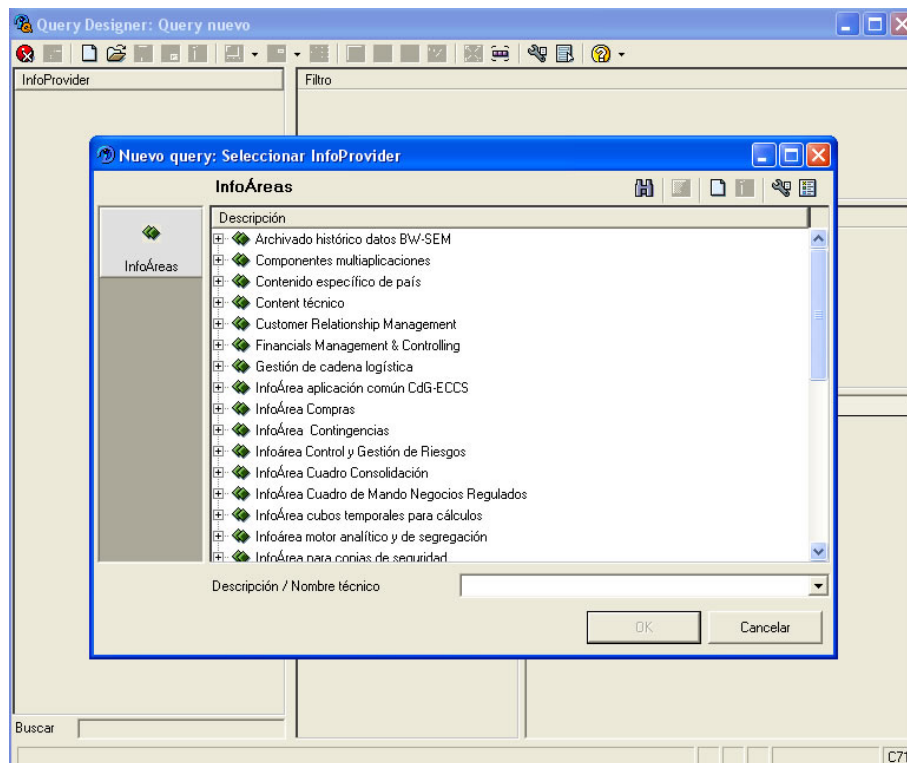



Figura 25: Query Designer: Crear query; seleccionar InfoProvider

Hay cierto número de ajustes que se pueden realizar y que afectan a la presentación y/o al comportamiento de la query, éstas son llamadas propiedades

de la query. Se accede mediante el icono .

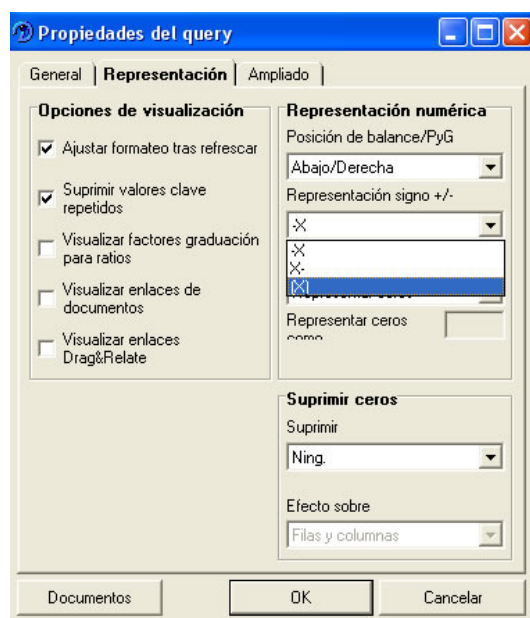


Figura 26: Query Designer: Propiedades del query

El principal cambio que hemos hecho en las propiedades es la representación de los números negativos, como mostramos en la imagen anterior.

Una vez cambiadas las propiedades, procederemos a crear la query. Inicialmente el programa muestra la siguiente apariencia:

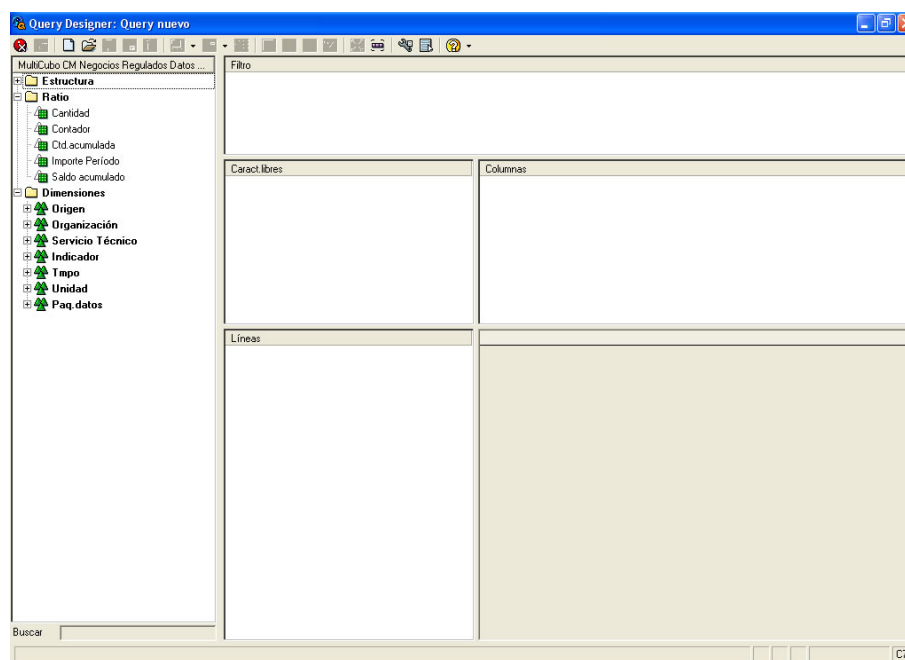


Figura 27: Query Designer: Pantalla inicial

Ahora, nuestra tarea será completar correctamente las áreas de trabajo definidas anteriormente.

Para crear un conjunto de líneas o de columnas, podemos crearlos desde cero o usar una estructura ya creada en otra query.

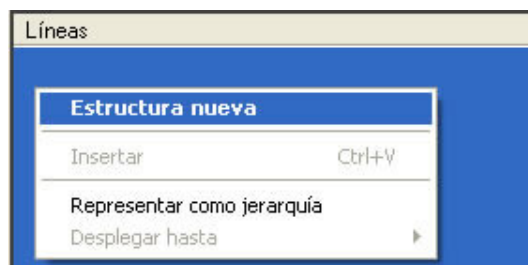


Figura 28: Query Designer: Crear una estructura nueva

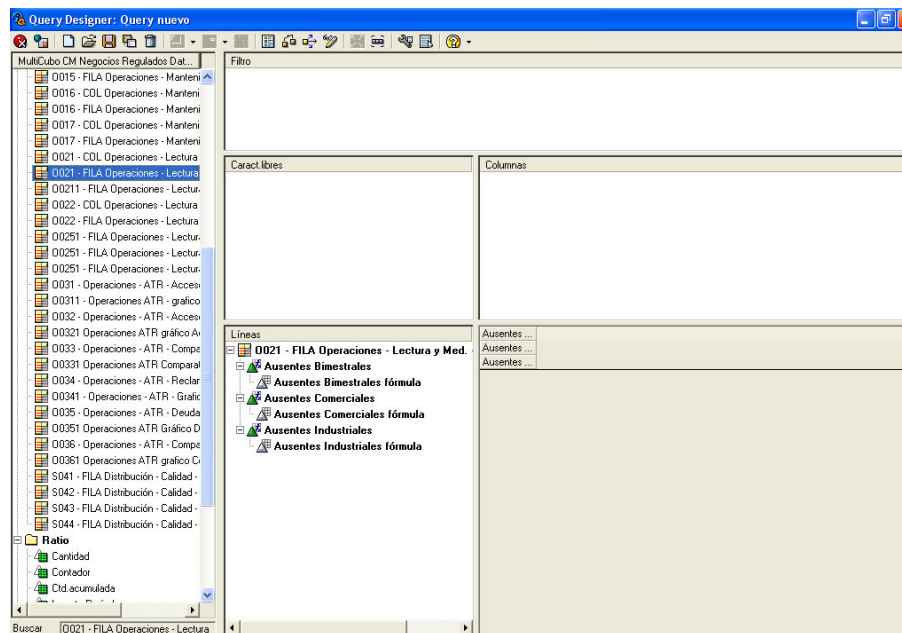


Figura 29: Query Designer: Usar estructura existente

Como ejemplo, vamos a crear desde el principio, una estructura que usaremos para las columnas de la tabla.

Una vez creada la estructura, situándonos sobre esta y pulsando el botón derecho, vemos que podemos crear una selección o una fórmula nueva. En la selección, indicaremos las características que mostrará esa fila /columna, ya sean características de tiempo, el concepto que representa o el sistema origen de donde obtiene los datos; generalmente, las selecciones las ocultaremos y no se mostrarán por pantalla. Por otro lado, las fórmulas, las usamos para operar con el resultado obtenido de las selecciones y representar sumas entre selecciones, porcentajes o el resultado sin operar obtenido en la propia selección.

Empezaremos creando una selección en la estructura de la columna y nos mostrará la siguiente pantalla.

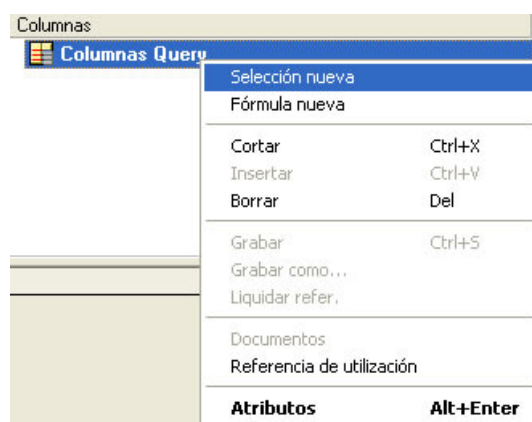


Figura 30: Query Designer: Columna, crear una nueva selección

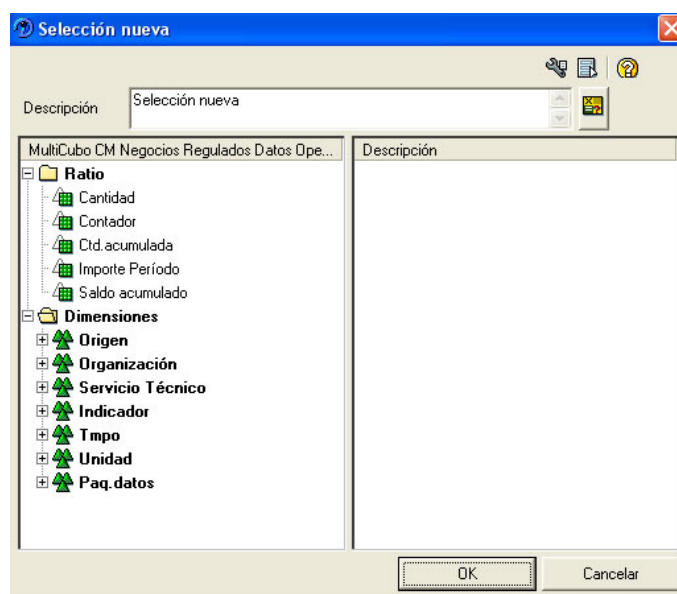


Figura 31: Query Designer: Columna, selección nueva

Una vez, tenemos creada la selección, le daremos un nombre para reconocerla después y usarla correctamente en las fórmulas en la que sea necesaria utilizarla.

Asimismo, procederemos a asignarle las características correspondientes para representar lo que necesitamos.

Para fijar un ratio o una dimensión a la selección, basta con arrastrarla del menú de la izquierda al campo en blanco de la derecha. Una vez allí, haciendo doble clic sobre la dimensión, podemos escoger la propiedad o característica que representará o incluso podemos excluir la característica que no queremos que quede mostrada en la query. También, podemos escoger un ámbito de valores para la consulta.

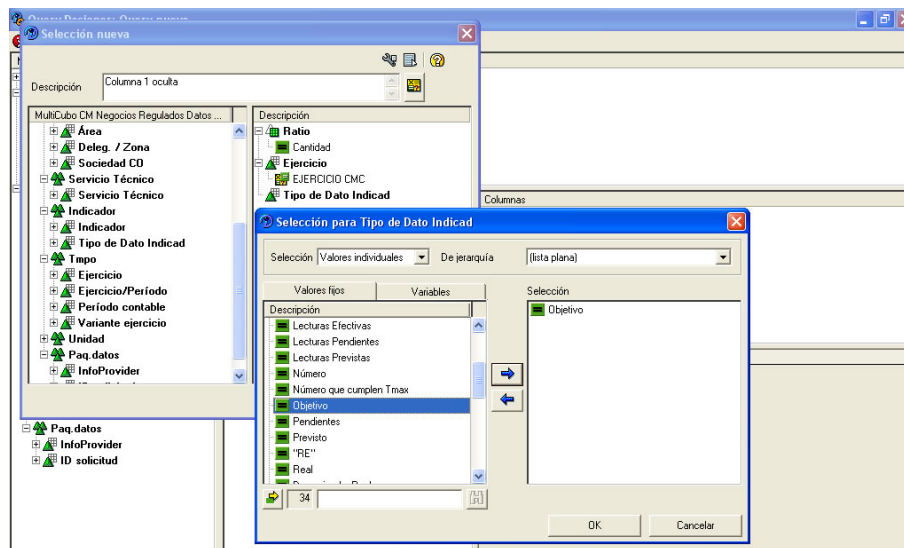


Figura 32: Query Designer: Trabajar con las dimensiones de la selección

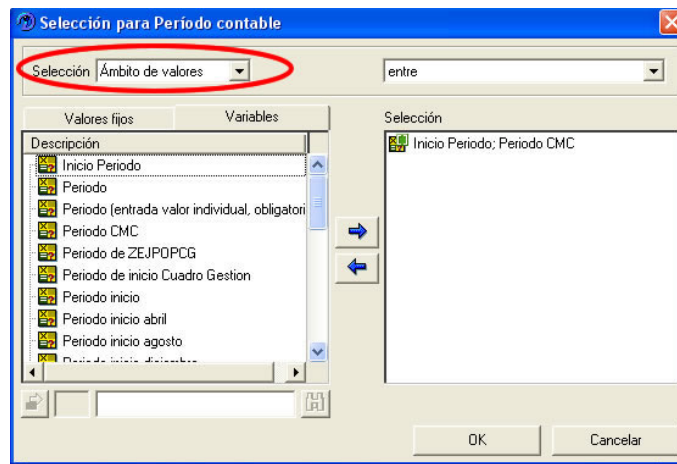


Figura 33: Query Designer: Ámbito de valores

Como hemos comentado antes, las selecciones las ocultaremos para que no queden visibles al usuario final. Ya que generalmente se usarán para crear fórmulas que si podrá ver el usuario final. Para ocultarlas, solo tenemos que seleccionarla, y en el menú que se despliega al hacer clic sobre el botón derecho, escoger atributos. Una vez se nos ha abierto la pantalla de propiedades, deberemos seleccionar en la ventana de suprimir, la opción de “siempre suprimir”

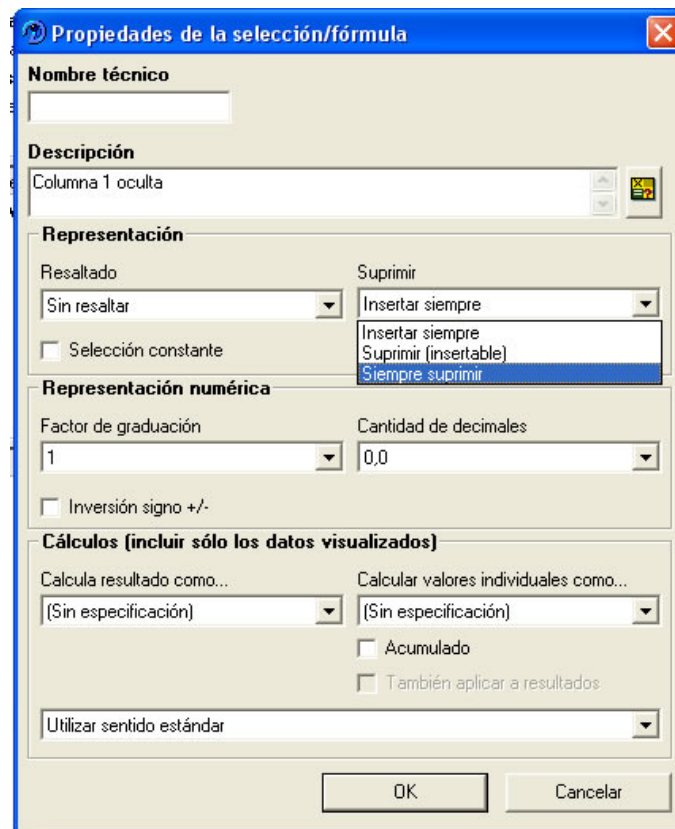


Figura 34: Query Designer: Ocultar selección

Una vez hemos creado la selección, procederemos a crear una fórmula con la que utilizar dicha selección.

Como primer ejemplo, crearemos una fórmula simple, que mostrará el valor obtenido en la selección. La función de datos NODIM, se usa para fórmulas en las que operemos con datos que tienen diferente unidad, (por poner un ejemplo, metros y segundos). Aunque en este caso sea innecesario, tomamos como costumbre usar siempre esta función.

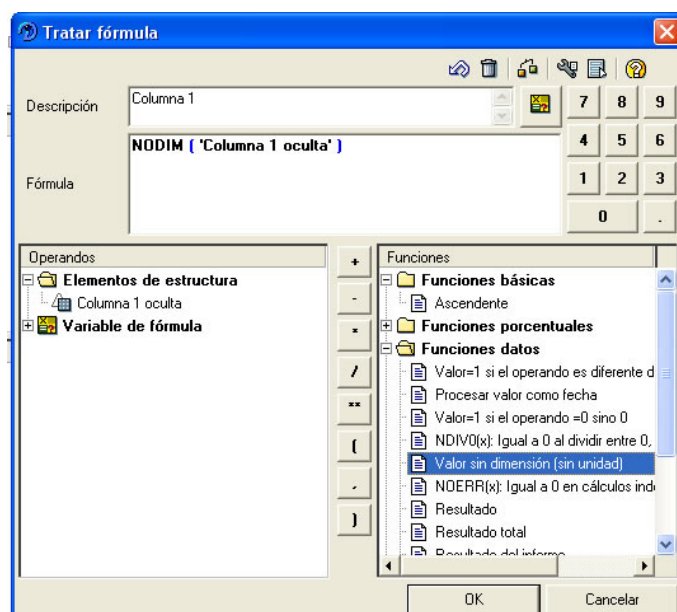


Figura 35: Query Designer: Fórmula sencilla

Ahora vamos a ver un ejemplo de fórmula que usa varias selecciones y en este caso si será útil y necesaria la función NODIM.

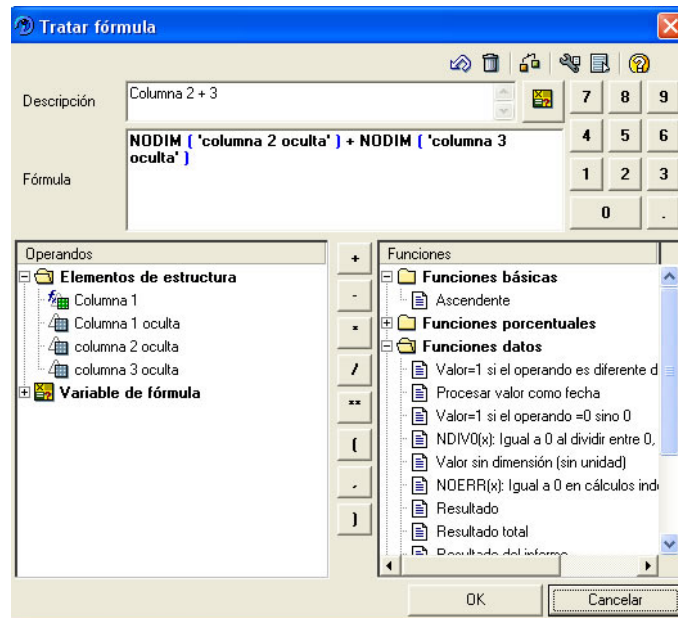


Figura 36: Query Designer: Fórmula de suma entre dos selecciones

En las fórmulas a parte de funciones aritméticas básicas como pueden ser la suma, resta, multiplicación o división, también podemos usar otras funciones matemáticas como logaritmos, funciones lógicas (and, or...) y funciones trigonométricas.

Los ratios solo pueden estar definidos en las filas o en las columnas, pero no en los dos sitios. Es decir, si usamos la propiedad ratio para definir las columnas, cuando creamos las filas, no podremos usar los ratios, y las celdas, dependerán del ratio que hallamos seleccionado para la columna correspondiente. Como podemos ver en la siguiente figura, no podemos escoger ratio, tan solo las dimensiones que definirán la selección.

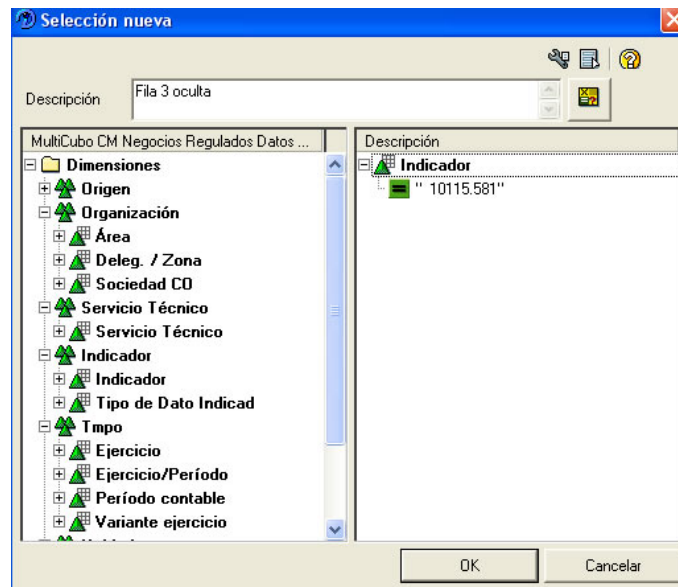


Figura 37: Query Designer: Selección sin ratio disponible

Excepciones:

Para marcar los valores de salida de los ratios con niveles diferenciadores podemos usar esta herramienta del diseño de queries, podemos definir una variable para introducir los valores en cada ejecución y modificarlos durante el análisis.

En el ejemplo que muestra la figura, para la columna, R/Ppto, hemos definido una excepción, que según el valor que se obtenga en la query, si está entre los límites de cada valor, se mostrará con un fondo de celda de un color u otro.

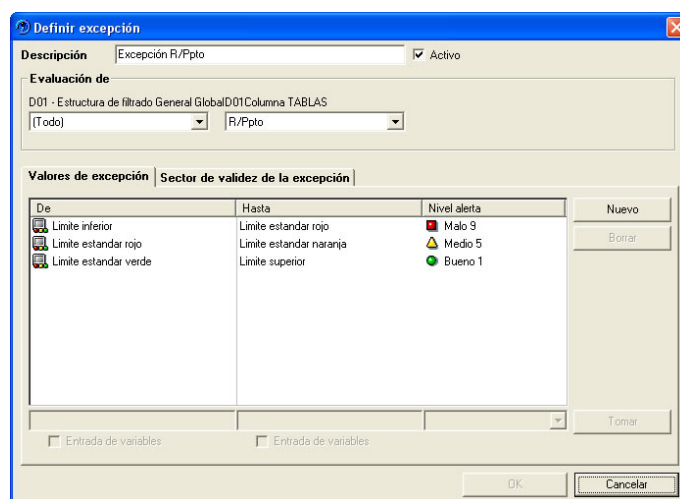



Figura 38: Query Designer: Diseño de excepciones

Los datos que cumplan estas excepciones aparecerán en las pantallas finales con estos colores diferenciadores para hacer más sencilla la comprensión de los datos por los usuarios finales.

Resumen Económico (M\$)	Real	Ppto	Aant	R/Ppto
Ingresos	14	0		X
Ventas				0,0
Otros ingresos Operativos	14	0		X
Gastos	7.546	7.104	5.746	6,2
Compras				0,0
Gastos de Personal	1.022	974	875	4,9
Tributos				0,0
Trabajos, suministros y servicios ext.	6.519	6.123	4.865	6,5
Otros gastos de explotación	5	7	6	(25,0)
Inversión	493	1.494	321	(67,0)
Total Redes				0,0
Total Acometidas				0,0
Construcción de plantas				0,0
Construcción E.R. & S.				0,0
Instalaciones Auxiliares de red	32	267	4	(88,0)
Otras inversiones	461	1.227	317	(62,5)

Figura 39: Query Designer: Tabla final con excepciones

Definición celda a celda:

El editor de celdas (botón ) sirve para dar una definición directa de celdas a una celda en particular. A su vez, también permite marcar una celda específica como “celda de referencia”, de forma tal que pueda ser referenciada como tal en futuros cálculos.

Las celdas son la intersección de 2 estructuras en una definición de consulta entonces el editor de celdas sólo puede ser utilizado en definiciones de consultas donde existan 2 estructuras. Una estructura puede contener sólo características mientras otra estructura puede contener sólo ratios.

Las celdas pueden ser fórmulas, selecciones, referencias o celdas auxiliares.

El editor de celdas es una función del BEx Query Designer. Se puede seleccionar el editor sólo si la definición de la consulta contiene 2 estructuras. Se puede definir un máximo de 2 estructuras en la definición de una consulta.

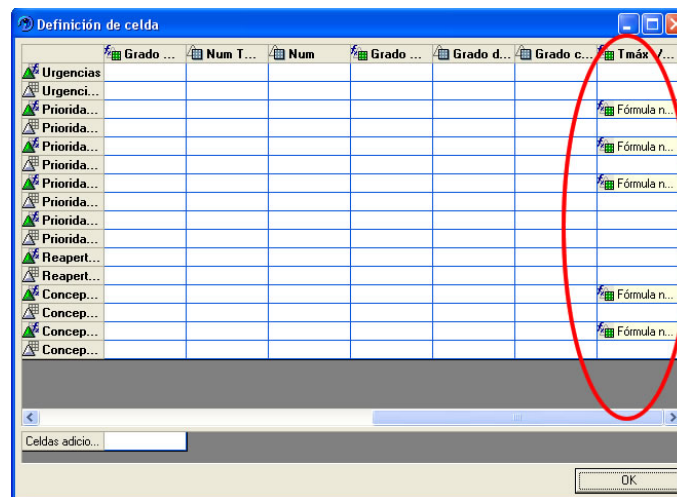


Figura 40: Query Designer: Definición celda a celda

La query que hemos ido creando paso a paso en este capítulo quedaría de esta forma en el query designer.

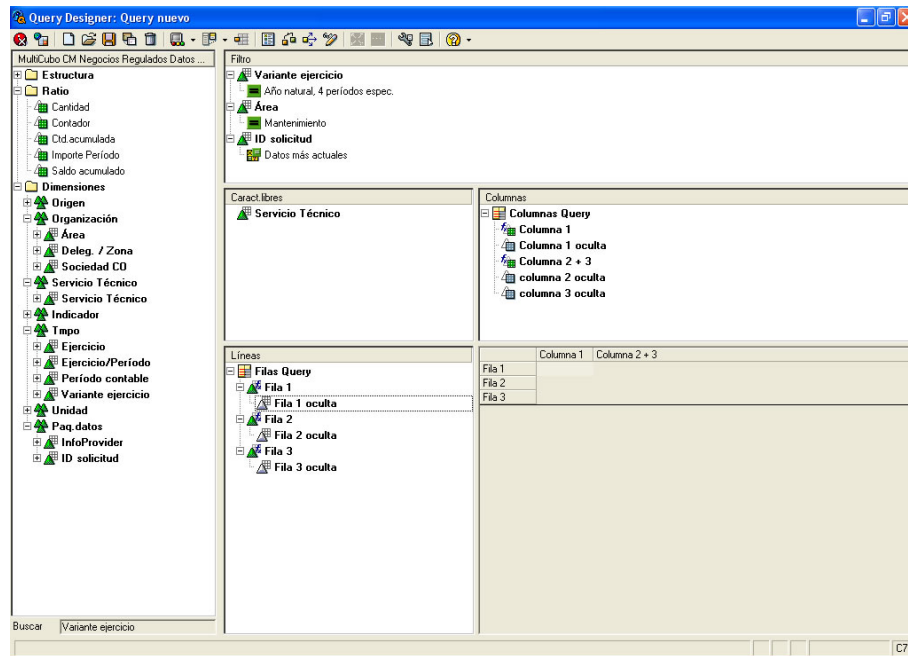


Figura 41: Query Designer: Query de ejemplo finalizada

Si queremos crear una query para un gráfico, deberemos cambiar el área de columnas, e incluir en dicha área el periodo que queremos que abarque el gráfico.



Figura 42: Query Designer: Área de columnas de una query de gráfico

En el área de línea, a parte de las filas que mostramos en la tabla, también incluiremos las columnas de la query de la tabla correspondiente. Con esta disposición, tendremos tantos gráficos diferentes como filas haya y tantas series de datos en el gráfico como columnas visibles tengamos en la query.

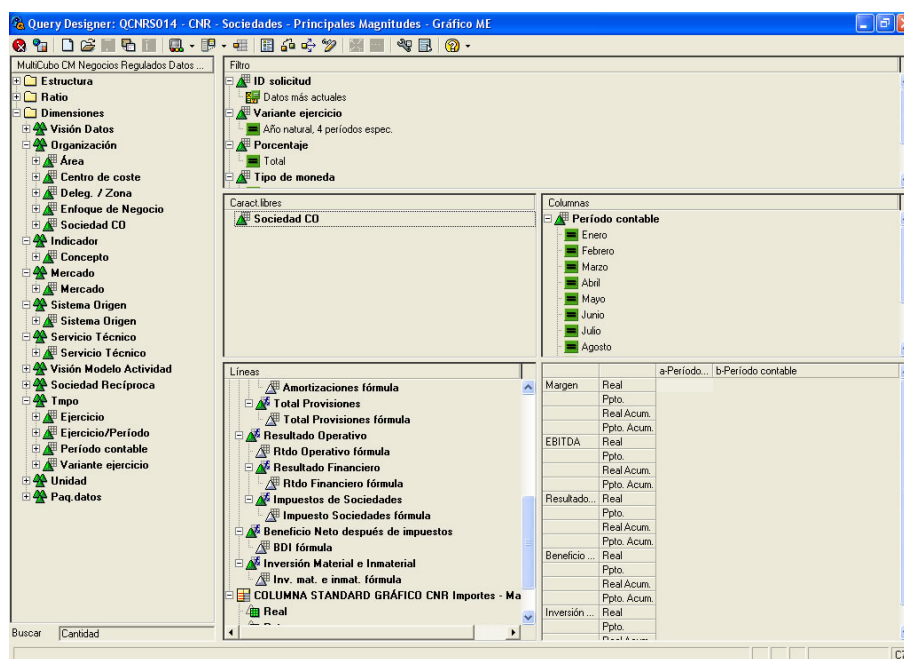


Figura 43: Query Designer: Ejemplo de una query de gráfico

Para finalizar el apartado correspondiente a este capítulo, mostraremos como ejemplo una query de tabla y una de gráfico y como quedarían finalmente en la aplicación final.

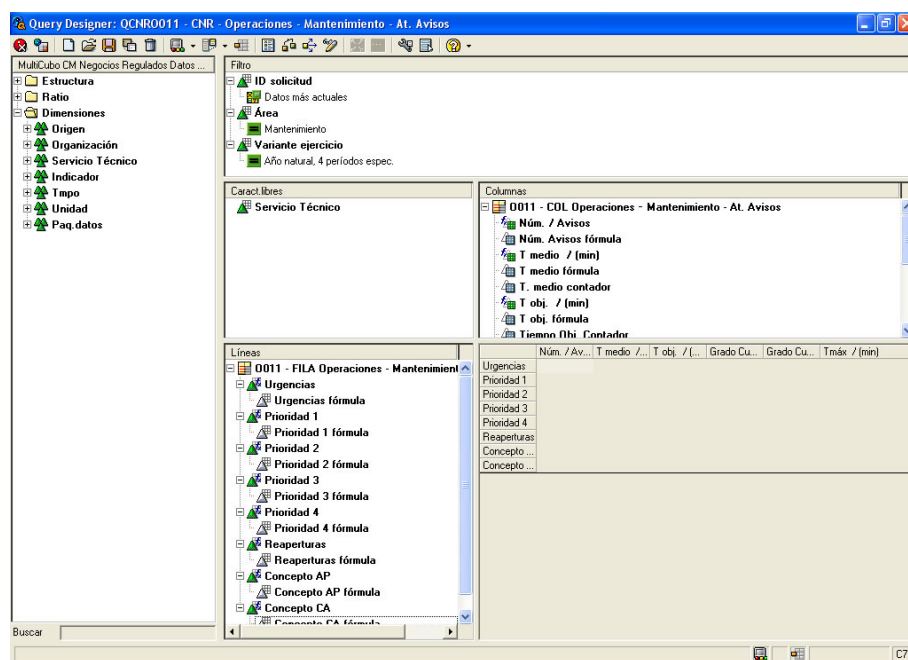


Figura 44: Ejemplo query tabla

Atención Avisos	Núm. Avisos	T medio (min)	T obj. (min)	Grado Cum Real (%)	Grado Cum Obj. (%)	Tmáx (min)
Urgencias						
Prioridad 1	4.592	25,8	0	97,2		60
Prioridad 2	13.315	37,0	0	99,2		120
Prioridad 3	16.927	95,4	0	99,0		480
Prioridad 4	3.443	83,2	N/A	N/A	N/A	N/A
Reaperturas						
Concepto AP	2.052	89,1	0	88,1		180
Concepto CA	2.039	80,9	0	81,2		180

Figura 45: Query tabla: Vista final

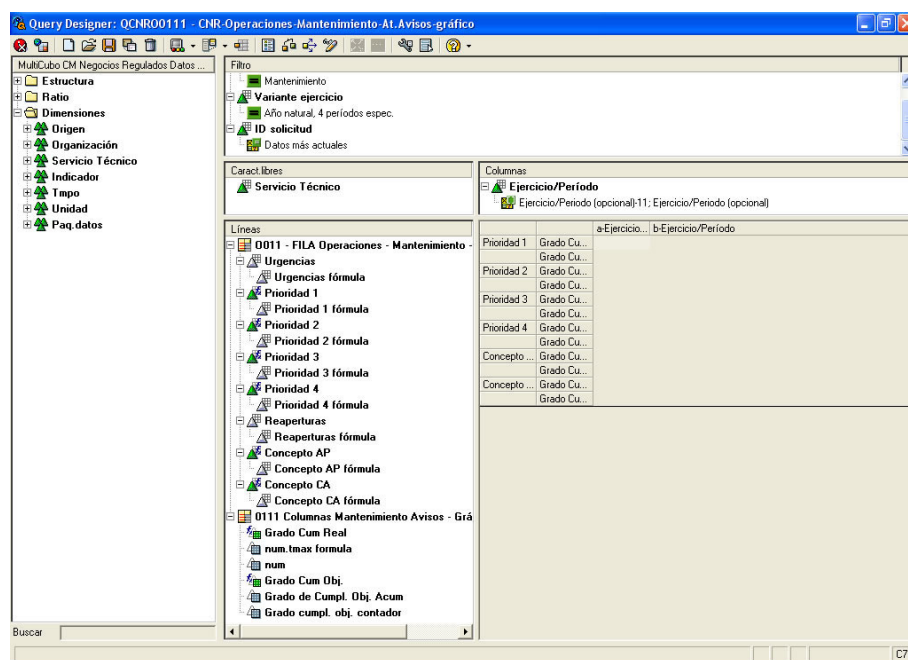


Figura 46: Ejemplo query gráfico

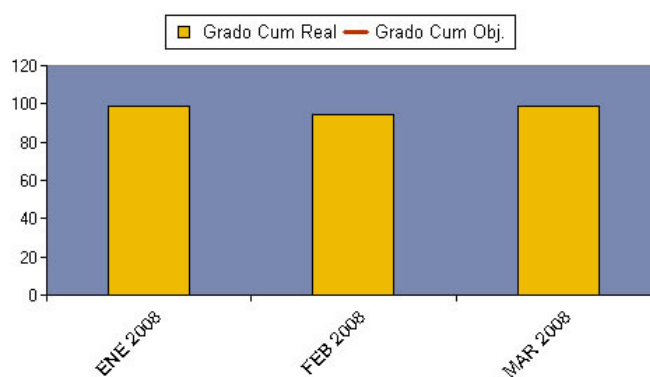


Figura 47: Query gráfico: Vista final

6.2.3. Web Application Designer (WAD)

BEx Web Application Designer es una aplicación de escritorio para crear utilidades Web con contenido específico para SAP BW. Usando BEx Web Application Designer, podemos crear una página de HTML que contiene contenido como varias tablas, gráficos o selectores. Esta página de HTML proporciona la base no sólo para usos Web con interacción compleja, sino que también para cuadros de mando de Web e iViews.

La pantalla principal del programa, se divide en tres subpantallas:

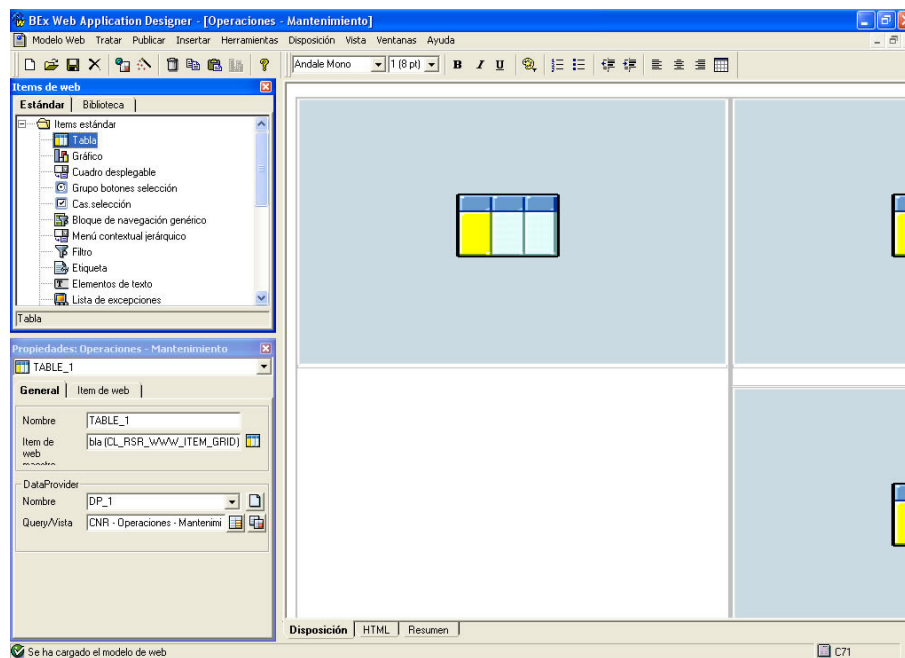


Figura 48: WAD: Pantalla general

Item de Web:

Desde esta subpantalla podemos añadir objetos a la pantalla final, tales como tablas, gráficos, filtros..., con los cuales mostraremos los datos de un data provider.

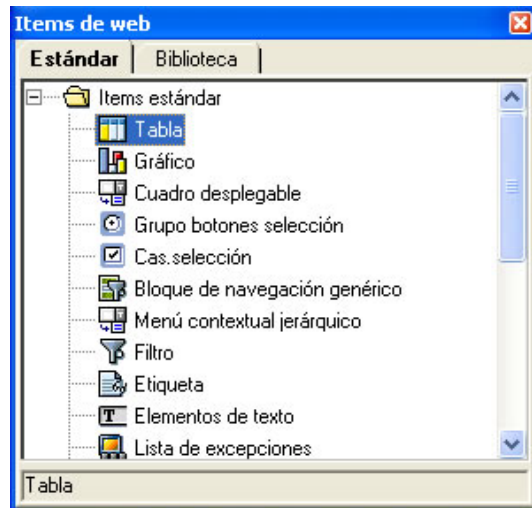


Figura 49: WAD: Subpantalla Item de web

Propiedades:

Podemos modificar las principales propiedades de la pantalla general, como el nombre técnico o la descripción.

Una vez tenemos seleccionado un gráfico, tabla o cualquier elemento de la pantalla, podremos modificar las propiedades de formato de este objeto.



Figura 50: WAD: Subpantalla Propiedades

Pantalla principal:

Contiene las plantillas Web que editamos en el proceso de diseño y que forman la base de nuestras aplicaciones Web.

En la ventana principal, tenemos 3 pestañas en la parte inferior:

La primera nos muestra la disposición de la página con los elementos que hemos incluido en la misma.

En la segunda tenemos el código HTML. Cada elemento de sintaxis HTML tiene un color diferente.

Elemento	Color
Comentarios	Gris
Código HTML	Azul
Constantes	Negro
Contenidos específicos de SAP BW	Rojo

Cuadro 2: WAD: colores para diferenciar código HTML

De todos modos, siempre podemos modificar el código HTML desde un editor externo que nos muestre de una manera más clara y concisa el contenido de la página.

En la pestaña resumen, vemos todos los objetos de la pantalla en un listado que nos muestra el tipo de ítem de web utilizado, el data provider del que muestra datos, y la query utilizada para hacer la selección de datos.

Item de web	Item de web maestro	DataProvider	Query/Vista
TABLE_1	Tabla	DP_1	CNR - Operaciones - Mantenimiento - At. Avisos
TABLE_2	Tabla	DP_2	CNR - Operaciones - Mantenimiento - Parametros de Calidad
TABLE_3	Tabla	DP_3	CNR - Operaciones - Mantenimiento - Reclamaciones
TABLE_4	Tabla	DP_4	CNR - Operaciones - Mantenimiento - Inspección Periodica
TABLE_5	Tabla	DP_5	CNR - Operaciones - Mantenimiento - Fugas Eliminadas (Tmax)
TABLE_6	Tabla	DP_6	CNR - Operaciones - Mantenimiento - Válvulas Seccionamiento
TABLE_7	Tabla	DP_7	CNR - Operaciones - Mantenimiento - Reseguimiento
DROPDOWNBOX_1	Cuadro desplegable	DP_11	CNR-Operaciones-Mantenimiento-At.Avisos-gráfico
DROPDOWNBOX_9	Cuadro desplegable	DP_12	CNR-Operaciones-Mantenimiento-At.Avisos-radar
DROPDOWNBOX_2	Cuadro desplegable	DP_21	CNR-Operaciones-Mantenimiento-Param.Calidad-gráfico
DROPDOWNBOX_10	Cuadro desplegable	DP_22	CNR-Operaciones-Mto-Param.Calidad-Radar Sat
DROPDOWNBOX_11	Cuadro desplegable	DP_23	CNR-Operaciones-Mto-Param.Calidad-Radar Insat
DROPDOWNBOX_3	Cuadro desplegable	DP_31	CNR-Operaciones-Mantenimiento-Reclamaciones-gráfico
DROPDOWNBOX_4	Cuadro desplegable	DP_41	CNR-Operaciones-Mantenimiento-Insp.Periódicas-gráfico
DROPDOWNBOX_5	Cuadro desplegable	DP_51	CNR-Operaciones-Mantenimiento-Fugas Eliminadas-gráfico
DROPDOWNBOX_6	Cuadro desplegable	DP_61	CNR-Operaciones-Mantenimiento-Valvulas Selecc-gráfico
DROPDOWNBOX_7	Cuadro desplegable	DP_71	CNR-Operaciones-Mantenimiento-Reseguimiento-gráfico
DROPDOWNBOX_8	Cuadro desplegable	DP_52	CNR-Operaciones-Mantenimiento-Fugas Eliminadas-radar
CHART_1: Tipo mixto	Gráfico	DP_11	CNR-Operaciones-Mantenimiento-At.Avisos-gráfico
CHART_2: Tipo mixto	Gráfico	DP_21	CNR-Operaciones-Mantenimiento-Param.Calidad-gráfico
CHART_3: Tipo mixto	Gráfico	DP_31	CNR-Operaciones-Mantenimiento-Reclamaciones-gráfico
CHART_4: Tipo mixto	Gráfico	DP_41	CNR-Operaciones-Mantenimiento-Insp.Periódicas-gráfico
CHART_5: Tipo mixto	Gráfico	DP_51	CNR-Operaciones-Mantenimiento-Fugas Eliminadas-gráfico
CHART_6: Tipo mixto	Gráfico	DP_61	CNR-Operaciones-Mantenimiento-Valvulas Selecc-gráfico
CHART_7: Tipo mixto	Gráfico	DP_71	CNR-Operaciones-Mantenimiento-Reseguimiento-gráfico
CHART_8: Radar	Gráfico	DP_52	CNR-Operaciones-Mantenimiento-Fugas Eliminadas-radar
CHART_9: Radar	Gráfico	DP_12	CNR-Operaciones-Mantenimiento-At.Avisos-radar
CHART_10: Radar	Gráfico	DP_22	CNR-Operaciones-Mto-Param.Calidad-Radar Sat
CHART_11: Radar	Gráfico	DP_23	CNR-Operaciones-Mto-Param.Calidad-Radar Insat

Figura 51: WAD: Pantalla principal, pestaña Resumen

Una vez estamos en esta pestaña de resumen, al seleccionar un elemento, la subpantalla de propiedades nos muestras las característica de este elemento, las cuales podremos modificar desde aquí.

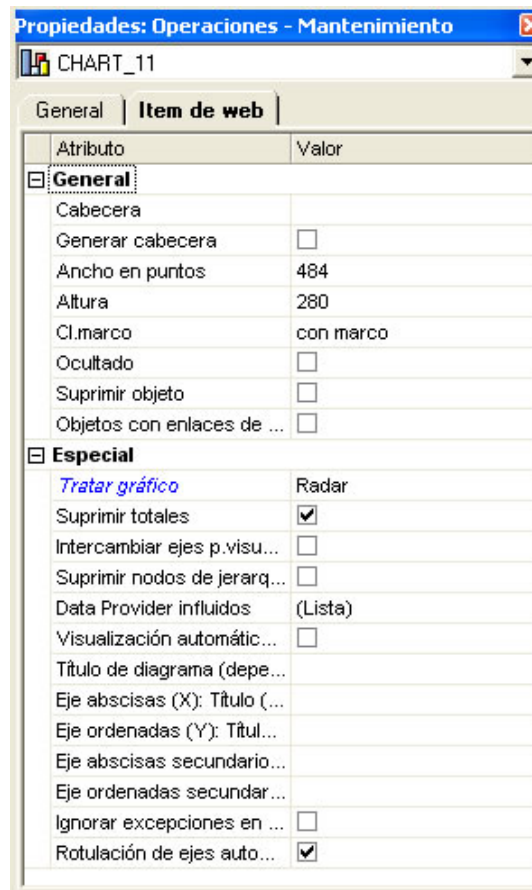


Figura 52: WAD: Propiedades del gráfico

Generalmente, desde aquí modificaremos el filtro de los selectores y el tipo de gráfico a mostrar, puesto que las medidas de la tabla o la situación de las mismas, las imponemos desde el código HTML.

Como podemos ver en la siguiente imagen, un selector puede filtrar por la fila, por la columna o por los filtros definidos en la query que representa.

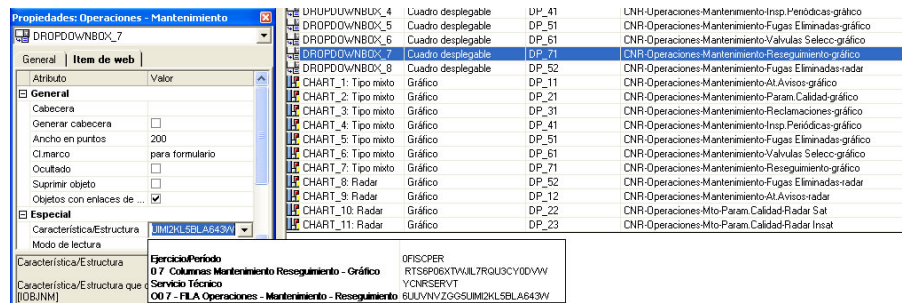


Figura 53: WAD: Elegir por que característica filtramos en un selector

Para modificar gráficos, seleccionamos uno y seleccionamos “tratar gráfico”, se nos abrirá una ventana con el editor de gráficos, similar al que podemos encontrar en el Microsoft Excel.

En este proyecto, hemos utilizado diferentes tipos de gráfico tales como gráficos de columnas, líneas, radares y mixtos entre los dos primeros.

El número de series de datos que seleccionamos, se corresponderá con el número de características diferentes que queramos mostrar en el gráfico.

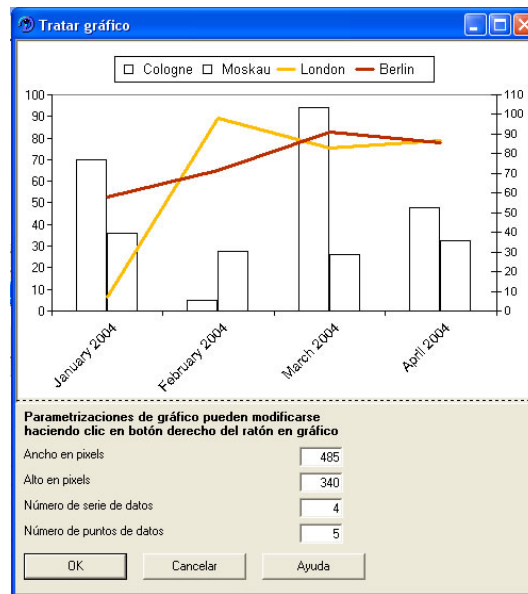


Figura 54: WAD: Tratar gráfico mixto

6.2.4. Business Explorer Analyzer

El BEX, es la herramienta que Business Warehouse pone a disposición del usuario final para el tratamiento de las consultas.

El BW utiliza el Excel como herramienta de reporting. Esto provoca que además de las funcionalidades básicas del Excel, se añada también una barra de herramientas propia del Bex Analyzer:



Figura 55: BEx: Barra de herramientas

El BW utiliza macros para poder generar sus informes sobre Excel. Por esta razón, el nivel de seguridad del programa deberá estar en nivel medio. Para comprobar este hecho se ha de proceder de la siguiente forma:

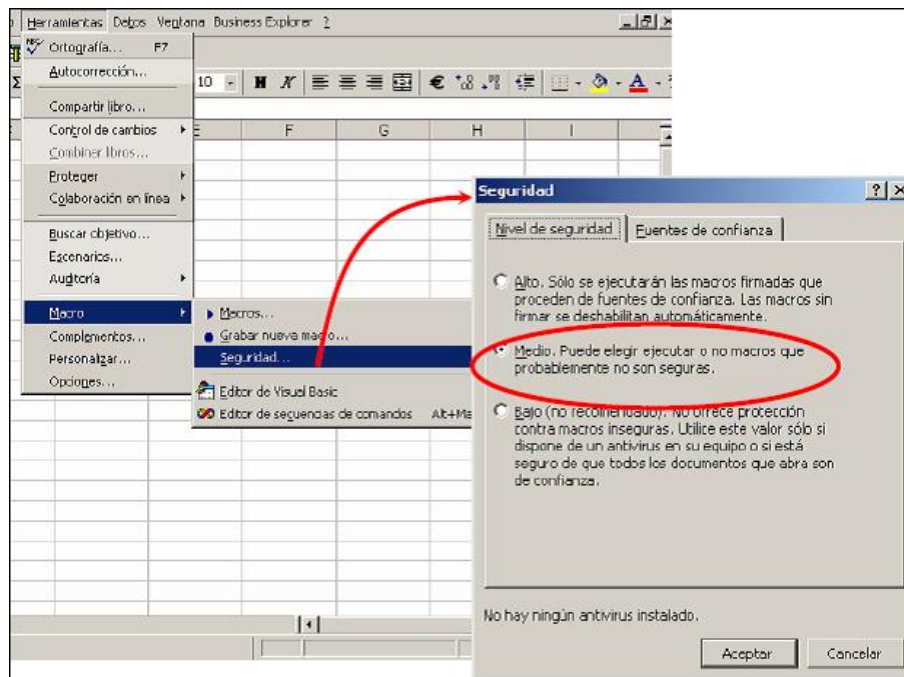


Figura 56: Bex: Habilitar macros

Al abrir una query, nos sale la siguiente pantalla para seleccionar las variables.

Variable	Valor
Periodo inicio	1
Periodo CMC	1
EJERCICIO CMC	2007
Tipo Dato (Opcional)	
Limite inferior	300-
Limite estandar rojo	10-
Limite estandar naranja	5,00000-
Limite estandar verde	10
Limite superior	300,00000
Soc. Italia Distribuidoras	0060
Soc. Italia Comercializadoras	0065

Figura 57: Bex: Abrir query

En la pantalla de selección existen dos tipos de variables: Variables Obligatorias: son de necesaria complementación para poder ejecutar el informe. La casilla asociada a estas variables viene informada con un valor por defecto o con el carácter '?'. Variables Opcionales: son de complementación opcional para poder ejecutar el informe. La casilla asociada a estas variables viene informada con un valor por defecto o con el carácter '?'.

7. Desarrollo e implementación

En este séptimo capítulo, destinado al desarrollo, mostraremos como hemos creado la aplicación utilizando las diferentes tecnologías que hemos explicado en el capítulo anterior.

Por motivos de confidencialidad y también para evitar cierta redundancia de información entre pantallas de corte similar, en esta parte del proyecto solo trabajaremos con cuatro pantallas y tras ver como se ha llevado a cabo el desarrollo, veremos estas pantallas concretas de la aplicación que vamos a desglosar paso a paso a continuación indicando que métodos hemos utilizado en cada parte de dichas pantallas.

7.1. Pantallas de navegación

Comenzaremos por el principio, por la pantalla de inicio de la aplicación, la cual tiene el siguiente aspecto.



Figura 58: Pantalla de Inicio

La pantalla de Inicio presenta las principales pantallas divididas por temática de forma amena e intuitiva al usuario.

La pantalla correspondiente al mapa del sitio, es bastante similar en cuanto a contenido y a tecnologías utilizadas, y en nuestra aplicación tendría el siguiente aspecto:

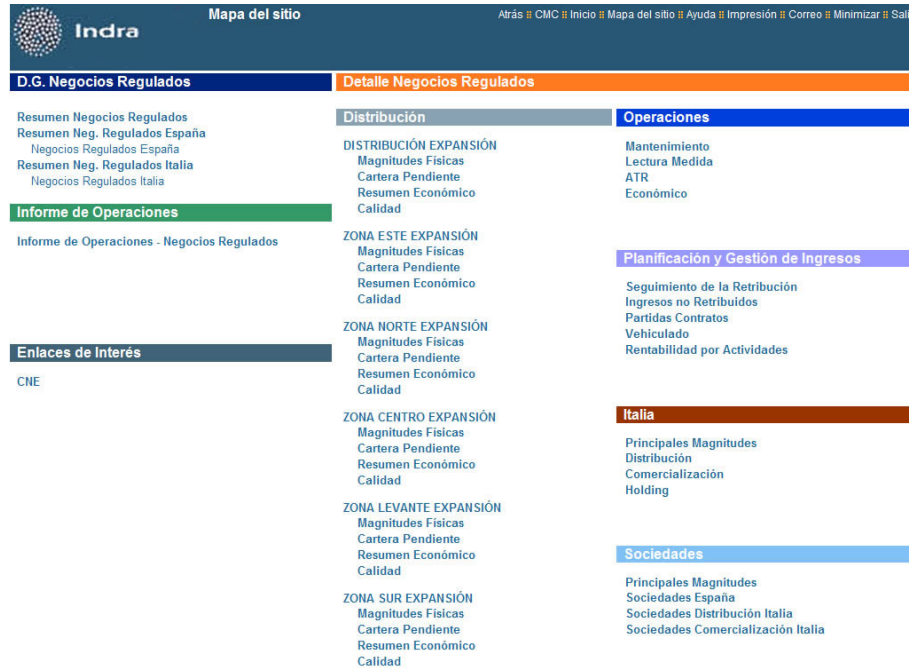


Figura 59: Mapa del Sitio

Esta segunda pantalla, al igual que la anterior, contiene todos los enlaces a las pantallas del CM NNRR divididos por temáticas y a las pantallas de resumen de negocios regulados de CMC.

En estas dos pantallas, no utilizaremos ninguna Query ni ningún objeto propio de SAP, hecho que solo sucederá en las pantallas de navegación, que como es de suponer, servirán para que el usuario llegue fácilmente a la información que desea visualizar.

El principal elemento de la pantalla inicial y del mapa del sitio es el diseño Web, puesto que simplemente hemos utilizado HTML y JavaScript para el desarrollo de ambas.

Todas las pantallas están hechas en HTML, e incluyen los scripts necesarios para la ejecución de la página entre las etiquetas `<head>` y `</head>`, que también contendrán el título de la pantalla.

```
<HEAD>
<META NAME="GENERATOR" Content="Microsoft DHTML Editing
Control">
<TITLE>Inicio CNR</TITLE>
<link href="/sap/bw/.../BWReports_jcm.css" type="text/css"
rel="stylesheet"/>
<script type="text/javascript"
src="/sap/bw/.../JavaScript/CM_CNR/CNR_contenidos.js">
</script>
<script>
getFicherosJs();
</script>
<script>
// resto de funciones JavaScript.
</script>
```

En el archivo CNR_contenidos.js tenemos las funciones JavaScript más utilizadas en las pantallas de la aplicación. Las cuales explicaremos en el apéndice (9.2).

El script `getFicherosJs` contenido en CNR_contenidos.js, llama a los archivos que contienen las funciones JavaScript que utilizaremos después, las cuales, como hemos comentado antes, las veremos con más detalle en el apéndice.

Una vez hemos definido todos los scripts necesarios, y cerrada la etiqueta `<head>`, procederemos a diseñar el cuerpo de la pantalla.

La pantalla principal consistirá en un marco superior que contendrá el símbolo de la compañía, la fecha, un enlace al mapa del sitio, otro al cuadro de mando CMC y dos otros dos para minimizar y salir de la aplicación.

El logo de la compañía, se ha insertado en el marco superior, como cualquier imagen en una pantalla hecha con el lenguaje de marcado HTML.

```
</img>
```

Para incluir la fecha, hemos utilizado el siguiente JavaScript:

```
<script type="text/javascript">document.write(getToday());</script>
```

Los enlaces al cuadro de mando CMC y el Mapa del Sitio se han creado de la siguiente manera:

```
<A href="JavaScript:irA(direccionDe('PCMC_NRCM'));" >CMC</A>
<A href="JavaScript:irA(direccionDe('PCNR000'));" >Mapa del sitio</A>
```

Por último, las acciones de minimizar y salir de la aplicación se han implementado con el siguiente código:

```
<A href="JavaScript:faMinimizar();" >Minimizar</A>
<A href="JavaScript:salirCMC()">Salir </A>
```

El marco principal, contendrá divididos en categorías, enlaces a todas las pantallas creadas para la aplicación y enlaces a páginas oficiales del sector. Asimismo habrá cuatro imágenes sobre el negocio y mercado que engloba la compañía, pero que hemos sustituido por los logos de Indra, SAP y la Universitat de Lleida para mantener en el anonimato a nuestro cliente.

Los submenús de enlaces, los hemos creado de dos formas diferentes. Los que aparecen a la izquierda de la aplicación son menús fijos, que los mostramos en pantalla con la función getEnlacesMap.

```
<script>getEnlacesMap(aEnlace['PCNRIO01'],'Informe de Operaciones');
</script>
```

Sin embargo, los menús de la parte derecha de la pantalla de inicio, son dinámicos y los podemos ocultar, quedando simplemente el título del grupo en un fondo naranja.

```
makeMenu('top','Operaciones')
makeMenu('sub','Mantenimiento',"aEnlace['PCNRO01']",'template')
makeMenu('sub','Lectura Medida',"aEnlace['PCNRO02']",'template')
makeMenu('sub','ATR',"aEnlace['PCNRO03']",'template')
makeMenu('sub','Econ&acute;mico',"aEnlace['PCNRO04']",'template')
```

Aunque se hayan construido de una forma diferente, ambos menús utilizan los enlaces definidos en el archivo CNR_enlaces.js. Los cuales contienen la información necesaria para enlazar con la pantalla requerida.

Por último, los enlaces de interés, se abrirán mediante una función propia de JavaScript en una ventana nueva.

```
<A href="JavaScript:open_w('http://www.webenlazada.es')">Web enlazada
</A>
```

En la pantalla correspondiente al Mapa del Sitio, tenemos la misma información, pero distribuida de una manera diferente como hemos podido anteriormente en este mismo capítulo.

No obstante, en Mapa del Sitio, los enlaces al resto de pantallas, son del mismo tipo:

```
<script>getEnlacesMap(aEnlace['PCNRIO01'],'Informe de Operaciones -
Negocios Regulados');</script>
```

```
<script>getEnlacesMap(aEnlace['PCNRO01'],'Mantenimiento');</script>
```

Los enlaces externos, tienen el mismo formato que en la pantalla de inicio descrita anteriormente.

Cabe destacar, que en esta pantalla, el marco superior contiene más enlaces, pero todos contruidos de la misma forma que hemos explicado en la pantalla de inicio.

7.2. Pantallas de información

Una vez hemos profundizado en el desarrollo de las pantallas de navegación, es el turno de explicar como hemos implementado las pantallas de información. Para ello, veremos la pantalla de Operaciones – Mantenimiento. Esta elección se debe a que es una página bastante completa, en la que tenemos numerosas tablas y todos los tipos de gráficos que nos podemos encontrar a lo largo del desarrollo de la aplicación.

Antes de explicar como crearemos la pantalla, veremos una muestra de la misma:

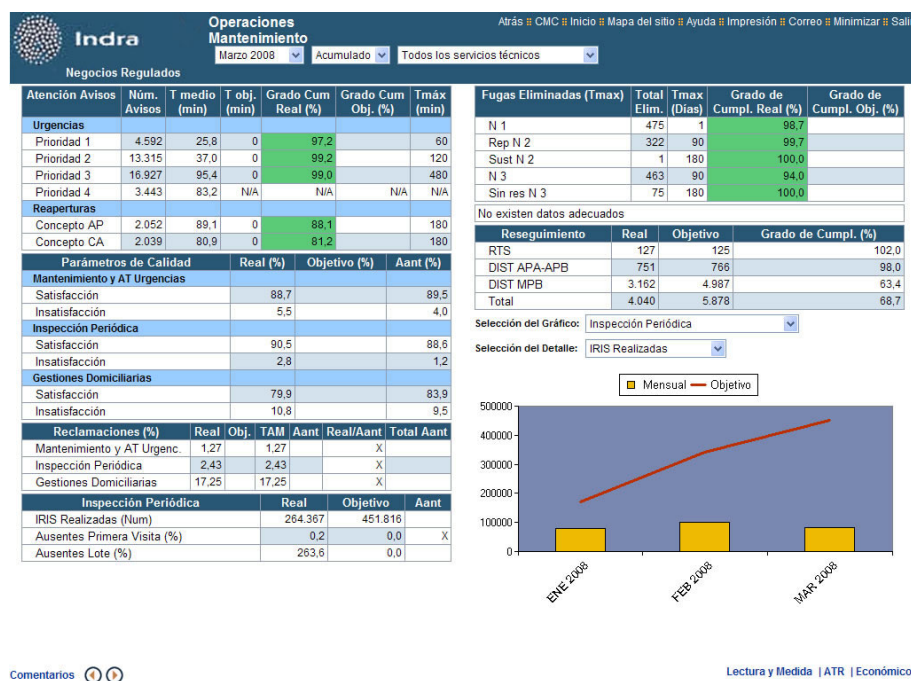


Figura 60: Pantalla Operaciones Mantenimiento

Esta pantalla está compuesta por siete tablas. Cada una de estas tablas puede tener asignados uno, dos o en algunos casos ningún gráfico. Tenga los gráficos que tenga cada pantalla, solo se mostrará uno, el escogido en el selector de

gráfico. Pero antes de hablar sobre esto, como hemos hecho en las pantallas anteriores, iniciaremos la explicación de la página con el marco superior de la misma.



Figura 61: Pantalla Operaciones Mantenimiento: Cabecera

Como sucedía en la pantalla inicial, en el marco superior, tenemos el logo de la empresa (aunque aparezca el logo de Indra por motivos de confidencialidad) en el margen izquierdo. En el centro tenemos una serie de selectores que son diferentes en cada pantalla y se utilizan para filtrar la información temporalmente por diversas características de filtrado. Una vez más, nos encontramos con una serie de enlaces en la parte derecha del margen superior.

Empezaremos comentando los enlaces del marco superior, los cuales no serán los mismos que en las pantallas de navegación descritas anteriormente, pero si que serán iguales entre todas las pantallas de información.

Junto a los links que enlazan a las ya conocidas pantallas de Mapa del Sitio e Inicio, desde estas pantallas de información también podremos acceder al cuadro de mando CMC, ver la ayuda de la aplicación o imprimir la pantalla entre otras utilidades. Además, a los links que teníamos en las pantallas de navegación de minimizar y salir, se une otro para volver a la pantalla que habíamos visitado con anterioridad.

En la parte central del marco superior, nos encontramos con los selectores desplegables de cabecera, los cuales afectan a toda la página y al contrario que los enlaces de este marco, cambian de una página a otra, según la necesidad de filtrado de cada una.



Figura 62: Pantalla Operaciones Mantenimiento: Filtro cabecera

En la pantalla en la que nos encontramos, incluiríamos los filtros así:

```
<script>
getPageHead('Operaciones*Mantenimiento','PERIODO_DATO-MACMC-
SERV_TEC');
</script>
```

El primer parámetro de la función es el título que tendrá la pantalla, siendo el símbolo *, un salto de línea. El segundo parámetro contiene el nombre de los filtros de cabecera que afectarán a la pantalla. Estos selectores se encuentran en el archivo CNR_contenidos.js, dentro de la función getPageHeadSelectors a la cual llamamos desde getPageHead.

Para una mejor comprensión, veremos una parte de esta larga función, que trata a todos los selectores que tenemos en la aplicación.

Dentro de la función getPageHeadSelectors y una vez definidas las variables, comprobamos si la pantalla contiene selectores.

En el caso que no haya ninguno:

```
if (aSelect[i]=="") strAux = strAux + "<br><br>";
//si no hay selectores dejo un espacio
```

Si por el contrario, tenemos uno o más selectores, que será el caso habitual, entrará en la condición que cumpla:

```
else if (aSelect[i]=="MACMC")
{
strAux = strAux +
getSelForm ("fMACMC", new Array ( new Array("Mensual","Mensual"), new
Array("Acumulado","Acumulado")),
getMACMC(),
"setEstado('fMA',this[this.selectedIndex].value);
setEstado('ZPERIN',this[this.selectedIndex].value);
irA(SAP_BW_URL_Get( )+getMenAcu(this[this.selectedIndex].value));");
}
```

Una vez expuesto como desarrollaríamos el marco superior de la pantalla, vayamos al marco central, en el que mostraremos los datos obtenidos con las consultas diseñadas con el Query Designer en forma de tabla o gráfico. Como comentábamos al principio de este apartado, la pantalla contiene siete tablas y un gráfico. Ahora pasaremos a explicar como las creamos y como las situamos dentro de la pantalla.

El primer caso que afrontaremos será crear una tabla tal y como nos la muestran la formulación y la maquetación de la que hemos hablado en el capítulo 4.

Si queremos obtener una tabla como esta:

Atención Avisos	Núm. Avisos	T medio (min)	T obj. (min)	Grado Cum Real (%)	Grado Cum Obj. (%)	Tmáx (min)
Urgencias						
Prioridad 1	4.592	25,8	0	97,2		60
Prioridad 2	13.315	37,0	0	99,2		120
Prioridad 3	16.927	95,4	0	99,0		480
Prioridad 4	3.443	83,2	N/A	N/A	N/A	N/A
Reaperturas						
Concepto AP	2.052	89,1	0	88,1		180
Concepto CA	2.039	80,9	0	81,2		180

Figura 63: Pantalla Operaciones Mantenimiento: Tabla Atención Avisos

Tendremos que seguir los siguientes pasos:

Primero tendremos que diseñar su correspondiente Query con el Query Designer. No vamos a crearla paso a paso, porque en el apartado correspondiente a este programa ya explicamos los pasos a seguir a la hora de crear una Query. Esta vez, la mostraremos ya hecha.

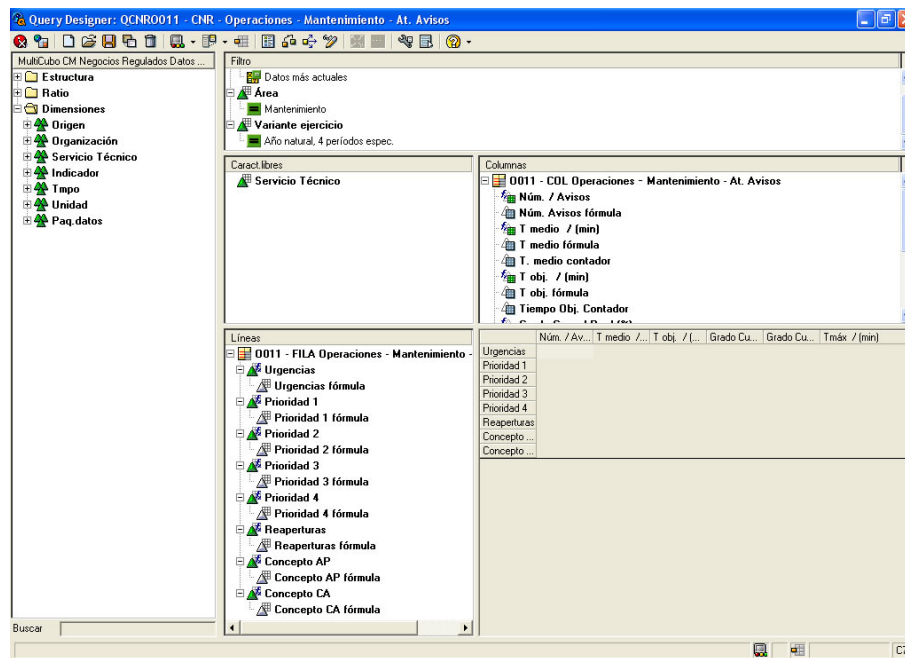


Figura 64: Query correspondiente a la tabla de Atención Avisos

Una vez tenemos hecha la Query correspondiente a la tabla deseada, tendremos que incluirla en la pantalla y situarla donde nos indica la maquetación en Power-Point.

Al empezar el código definimos los Data Providers (DP) que usaremos después en las tablas, gráficos y selectores. A dichos DP, les asignamos una Query y le indicamos en que InfoCubo está la citada Query.

```
<object>
<param name="OWNER" value="SAP_BW"/>
<param name="CMD" value="SET_DATA_PROVIDER"/>
<param name="NAME" value="DP_1"/>
<param name="QUERY" value="QCNRO011"/>
<param name="INFOCUBE" value="ICNRMOPER"/>
DATA_PROVIDER: DP_1
</object>
```

En este caso, el Data Provider, DP_1 utilizará la Query QCNRO011 del InfoCubo ICNRMOPER. De ahora en adelante, cuando necesitemos utilizar esta Query, nos referiremos a ella con el nombre del DP y no con el nombre técnico de la Query.

Una vez definida la tabla, tendremos que incluirla en su sitio correspondiente. Cuando hayamos localizado este sitio, introduciríamos el siguiente código:

```
<!--TABLA ATENCI&Oacute;N AVISOS-->
<object><param name="OWNER" value="SAP_BW"/>
<param name="CMD" value="GET_ITEM"/>
<param name="NAME" value="TABLE_1"/>
<param name="ITEM_CLASS" value="CL_RSR_WWW_ITEM_GRID"/>
<param name="DATA_PROVIDER" value="DP_1"/>
<param name="GENERATE_CAPTION" value=""/>
<param name="GENERATE_LINKS" value=""/>
<param name="WIDTH" value="485"/>
<param name="BLOCK_SIZE" value="50"/>
<param name="SHOW_PAGING_AREA_BOTTOM" value=""/>
<param name="TARGET_DATA_PROVIDER_1" value="DP_1"/>
<param name="MODIFY_CLASS"
value="ZCL_RSR_WWW_MODIFY_TABLE"/>
ITEM: TABLE_1
</object>
```

Ya hemos visto el ejemplo de cómo crear una tabla y situarla en la pantalla correspondiente. Ahora haremos lo propio con un gráfico.

Hemos escogido como ejemplo el gráfico de la tabla de Inspección Periódica de la pantalla que estamos utilizando para ver como se ha llevado a cabo la implementación de la aplicación.



Figura 65: Pantalla Operaciones Mantenimiento: Gráfico Inspección Periódica

Como sucede en el caso de las tablas, para el diseño del gráfico necesitamos usar el programa Query Designer con el cual diseñaremos una Query para dicho gráfico. Esta Query será similar a la de la tabla que corresponde al gráfico, pero como comentamos en el apartado 6.2.2., las columnas contendrán las características temporales y en las filas tendremos las dos estructuras utilizadas en la Query de la tabla.

Tras ver como crear la consulta necesaria, veamos una captura de cómo sería esta tras diseñarla con el Query Designer:

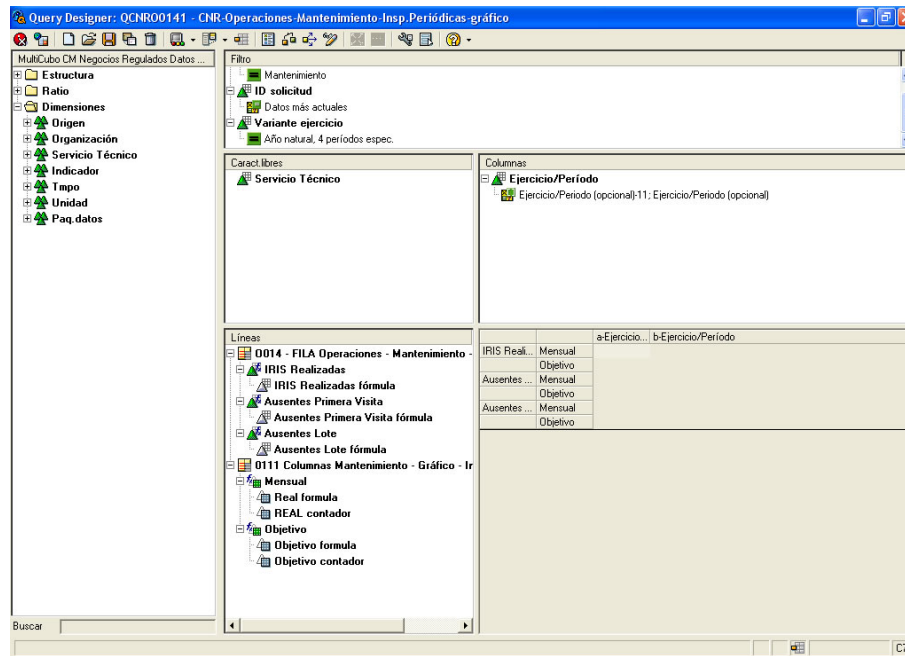


Figura 66: Query correspondiente al gráfico de Inspección Periódica

Para incluir este gráfico en la pantalla, actuaremos de la misma manera que con las tablas. Es decir, definiremos un DP nuevo, al cual asignaremos la Query recientemente creada y el InfoCubo sobre el que trabaja y a partir de este momento, cuando nos queramos referir a esta Query para utilizarla, la referenciaremos con el nombre del DP correspondiente.

```
<object>
<param name="OWNER" value="SAP_BW"/>
<param name="CMD" value="SET_DATA_PROVIDER"/>
<param name="NAME" value="DP_41"/>
<param name="QUERY" value="QCNRO0141"/>
<param name="INFOCUBE" value="ICNRMOPER"/>
DATA_PROVIDER: DP_41
</object>
```

En este caso, el Data Provider, DP_41 utilizará la Query QCNRO0141 del InfoCubo ICNRMOPER. De ahora en adelante, cuando necesitemos utilizar esta Query, nos referiremos a ella con el nombre del DP y no con el nombre técnico de la Query.

Una vez definido el gráfico, tendremos que colocarlo en su sitio correspondiente. Cuando hayamos localizado este sitio, introduciríamos el siguiente código:

```

<!--gráfico inspección periódica-->
<object><param name="OWNER" value="SAP_BW"/>
<param name="CMD" value="GET_ITEM"/>
<param name="NAME" value="CHART_4"/>
<param name="ITEM_CLASS"
value="CL_RSR_WWW_ITEM_CHART"/>
<param name="DATA_PROVIDER" value="DP_41"/>
<param name="GENERATE_CAPTION" value=""/>
<param name="GENERATE_LINKS" value=""/>
<param name="HEIGHT" value="280"/>
<param name="WIDTH" value="484"/>
<param name="BORDER_STYLE" value="BORDER"/>
<param name="TARGET_DATA_PROVIDER_1" value="DP_41"/>
<param name="AUTOMATIC_DESCRIPTION" value="X"/>
<param name="TMP_CHART_DATA_HANDLE"
value="IIP_5QR2L9PE6RCM75L74YGP7X3G"/>
ITEM: CHART_4
</object>

```

Después de ver como utilizamos las tablas y los gráficos, pasaremos a tratar un elemento no menos importante, los selectores de gráfico. En cada tabla que tengamos más de un gráfico, serán necesarios, puesto que el diseño técnico solo acepta un gráfico por pantalla, ocultando los demás. El gráfico que se mostrará será el que esté seleccionado en el selector. Asimismo, la mayoría de los selectores de gráficos tienen asociado un selector de detalle, que divide el gráfico en varios, pero mostrando solo uno, el que marca el selector.

Al contrario que con tablas y gráficos, los selectores no necesitan una Query o un DP nuevo, ya que utilizan el mismo DP que los gráficos.

No obstante, si que es necesario que lo situemos donde debe, con el siguiente código:

```

<object>
  <param name="OWNER" value="SAP_BW"/>
  <param name="CMD" value="GET_ITEM"/>
  <param name="NAME" value="DROPDOWNBOX_4"/>
  <param name="ITEM_CLASS"
value="CL_RSR_WWW_ITEM_FILTER_DDOWN"/>
  <param name="DATA_PROVIDER" value="DP_41"/>
  <param name="WIDTH" value="200"/>
  <param name="GENERATE_CAPTION" value=""/>
  <param name="IOBJNM" value="0H6UZ5Z62D2XUV36EY4WSII4S"/>
  <param name="SHOW_LABEL" value=""/>
  <param name="NO_REMOVE_FILTER" value="X"/>
  <param name="TARGET_DATA_PROVIDER_1" value="DP_41"/>
  <param name="HEIGHT" value="30"/>
  ITEM: DROPDOWNBOX_4
</object>

```

Cabe destacar dentro del código correspondiente al selector la siguiente línea:

```

<param name="IOBJNM" value="0H6UZ5Z62D2XUV36EY4WSII4S"/>

```

Este valor, es el correspondiente al filtro que utiliza el selector. En el caso que queramos filtrar por otra característica disponible en la Query, deberemos cambiar este código desde el Web Application Designer (6.2.3.).

Hasta ahora hemos creado todos los elementos que utilizaremos en la página. Es el momento de explicar como modificar las características de estos u ocultarlos.

Empezaremos explicando como dar formato a las tablas, que por defecto, aparecen sin título en la cabecera y con todas las filas compartiendo formato. Mediante una función JavaScript, podremos darle un nombre y resaltar filas en color para indicar que estamos hablando de totales o de títulos.

```

function iniPantalla( )
{
  //...
  //TABLA2
  var tbl = getTableByName('TABLE_2');
  if (tbl.rows.length>1)
  {
    formateoFilaTabla(tbl,0,'tblCabecera');
    formateoFilaTabla(tbl,1,'SAPBEXstdItemBC',0,true);
    formateoFilaTabla(tbl,4,'SAPBEXstdItemBC',0,true);
    formateoFilaTabla(tbl,7,'SAPBEXstdItemBC',0,true);
    cambiaTextCelda(tbl,0,0,"Parámetros de Calidad");
  }
  //...
}

```


La función condicional (`if (tbl.rows.length>1)`), la utilizamos en los casos que no tengamos datos, en esta situación, no se mostrará una tabla con cabecera y sin líneas debajo, sino que se mostrará un recuadro con el mensaje, “No existen datos adecuados”, como podemos ver en la figura 60. El resto del código sirve para dar formato de cabecera a la fila 0 y para resaltar el fondo de las filas 1, 4 y 7 que en esta tabla sirven como subtítulos dentro de la propia tabla. La última línea es para cambiar el contenido de la celda y que esta muestre el título de la tabla.

Tras finalizar todo lo que concierne a las tablas, pasaremos a explicar como ocultar los gráficos para que solo se muestre uno cada vez por pantalla y ocultar los selectores de detalle, en el caso que el gráfico no necesite un mayor nivel de detalle.

Por defecto, ocultamos todos los gráficos y selectores de detalle del gráfico y según la cookie elegida en el selector de gráfico, mostraremos unos u otros. En la función `iniPantalla()` es donde ocultamos todos los elementos mencionados.

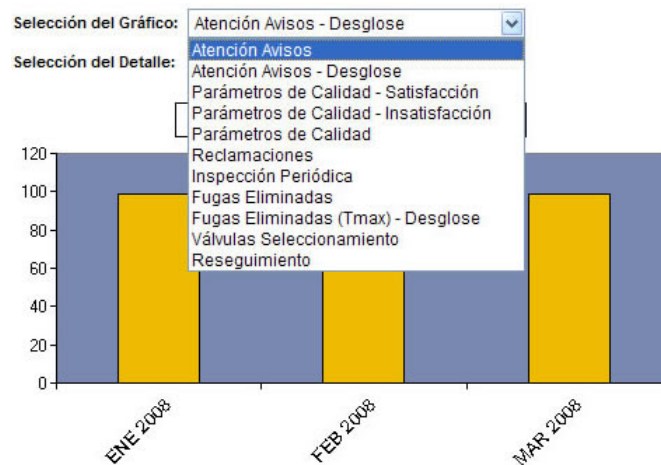


Figura 67: Selector del gráfico

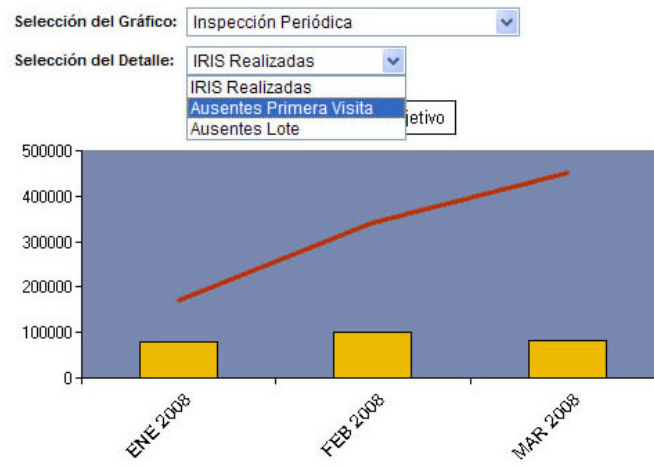


Figura 68: Selector del detalle

```
document.getElementById('table1').style.display = 'none';
document.getElementById('table2').style.display = 'none';
```

Donde table1 es el nombre que le damos al div del selector de detalle y table2 es como llamamos al div del gráfico correspondiente.

```
<div id=table1 style="OVERFLOW-Y: hidden; OVERFLOW-X: hidden;
HEIGHT: 25px">
<!--selector gráfico;fórmula atencióno;avisos desglose-->
<object>
<param name="OWNER" value="SAP_BW"/>
<param name="CMD" value="GET_ITEM"/>
<param name="NAME" value="DROPDOWNBOX_1"/>
<param name="ITEM_CLASS"
value="CL_RSR_WWW_ITEM_FILTER_DDOWN"/>
<param name="DATA_PROVIDER" value="DP_11"/>
<param name="WIDTH" value="200"/>
<param name="GENERATE_CAPTION" value=""/>
<param name="IOBJNM" value="4XTRDG8FZP4LILCP7FX28P5JG"/>
<param name="SHOW_LABEL" value=""/>
<param name="NO_REMOVE_FILTER" value="X"/>
<param name="TARGET_DATA_PROVIDER_1" value="DP_11"/>
<param name="HEIGHT" value="30"/>ITEM: DROPDOWNBOX_1
</object>
</div>
```

Una vez tenemos ocultos todos los div, con la función `iniMostrarOcultarDivs`, con el valor de la cookie almacenada, mostraremos el elemento correspondiente. Destacar que en el segundo caso ocultamos el selector de detalle por que la tabla correspondiente solo tiene una fila y no hace falta aumentar el nivel de detalle del gráfico.

```
function iniMostrarOcultarDivs(nom_cookie, cookie_defecto, id_sel)
{
  var valor_cookie = getEstado(nom_cookie, cookie_defecto);
  document.getElementById(id_sel).value = valor_cookie;
  switch(valor_cookie)
  {
    case "table1": document.getElementById('table1').style.display = "";
    document.getElementById('table2').style.display = "";
    break;
    //...
    case "table11": document.getElementById('table12').style.display = "";
    document.getElementById('detalle').style.display = 'none';
    //... }
  }
}
```

Cabe destacar que el selector de gráfico se mostrará siempre porque es un selector JavaScript, al contrario que el selector de detalle, que está hecho con objetos de SAP, puesto que utiliza la Query del gráfico a mostrar.

Desde el archivo JavaScript `CNR_enlaces.js`, podemos indicarle al selector de detalle, el valor que tomará por defecto y que por consiguiente será el gráfico que mostrará cuando escojamos el selector del gráfico deseado con la función `getURLCMDFilter`.

Para acabar la explicación de esta pantalla, nos centraremos en la parte inferior de esta. Es decir, en los enlaces de pie de página y el carrusel.

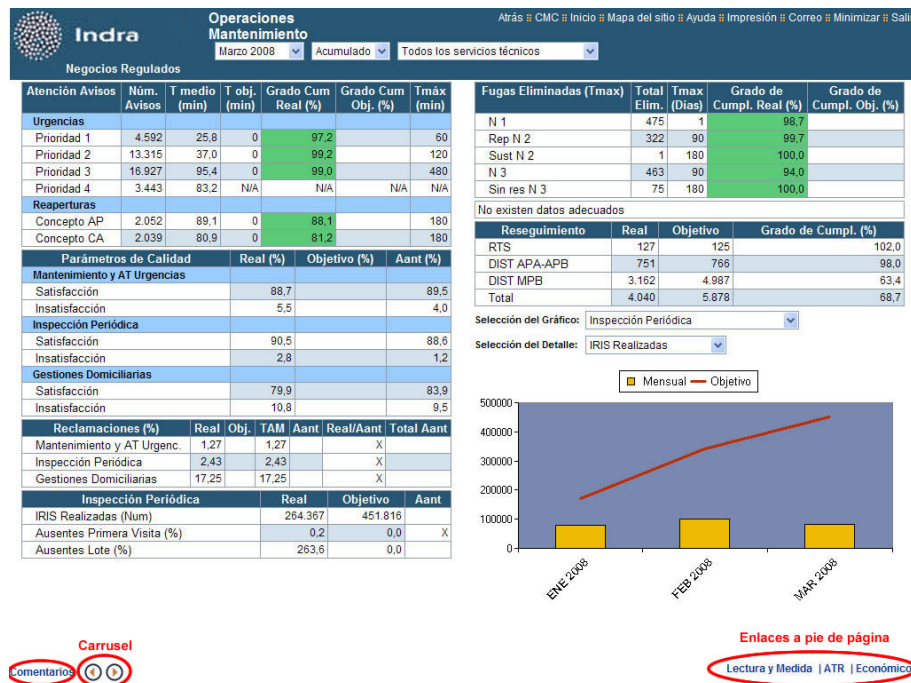


Figura 69: Comentarios, carrusel y enlaces de pie de página

Tras haber colocado todos los elementos en la pantalla, introduciremos el siguiente texto para incluir los ya nombrados, botón de comentarios, carrusel y enlaces de pie de página.

```
<script>
getPageFootLinks(SAPBWGetProperty('TEMPLATE_ID'));
getPageComent(SAPBWGetProperty('TEMPLATE_ID'),getPeriodo());
getImgNextPre(SAPBWGetProperty('TEMPLATE_ID'));
</script>
```

El botón de comentarios nos abrirá una nueva ventana y en ella aparecerán los documentos del mes de la pantalla que la ha cargado y que además sean los referidos a esa pantalla.

El carrusel se utiliza para navegar a la página anterior y siguiente respectivamente.

Las flechas se crean mediante la función `getImgNextPre()` que esta en el archivo `CNR_contenidos.js`, esta función crea un enlace a la pantalla siguiente y anterior tomando como defecto que la anterior es la misma pantalla pero con restándole

uno al último número y que la siguiente es sumándole uno al último número, así los links de PCNRI02 serían: Anterior PCNRI01, Siguiente PCNRI03.

Esto a veces no es posible usarlo y hay que indicar el valor de las pantallas posteriores y anteriores. Esto se indica en el archivo CNR_enlaces.js en la parte dedicada a excepciones de carrusel.

```
//- CARRUSEL -
//Carrusel (excepciones)
//Excepciones de distribución
aLinksNextPre['PCNRD00'] =
new Array(aEnlace['PCNRS04'],aEnlace['PCNRD01']);
```

Por último abordaremos los enlaces de pie de página, también llamados FootLinks.

Los footlinks son los links a otros cuadros de mando de la misma temática que el actual que aparecen en la parte inferior derecha de la pantalla.

La función que los crea es la getPageFootLinks() que esta declarada en el archivo CNR_contenidos.js

Esta función crea los footlinks a partir de los aLinksPaginas que aparecen en CNR_enlaces.js :

```
//- FOOTLINKS -
//Combox Italia(páginas italia) aLinksPaginas['PCNRI01'] =
new Array( new Array(aEnlace['PCNRI02'][0],
"Distribución", aEnlace['PCNRI02'][2]),
new Array(aEnlace['PCNRI04'][0],
"Comercialización",aEnlace['PCNRI04'][2]),
new Array(aEnlace['PCNRI05'][0], "Holding", aEnlace['PCNRI05'][2]) );
```

Pese al gran número de posibilidades que nos ofrecía la pantalla anterior, esta no usaba la función para ocultar columnas, puesto que siempre mostraba todas en cada una de las tablas que la componían. Para ver el funcionamiento de este script, hemos escogido la pantalla de Operaciones – Lectura y Medida, pero solo entraremos en detalle en el tema de la visibilidad de columnas en las tablas que así lo requiera el diseño técnico.

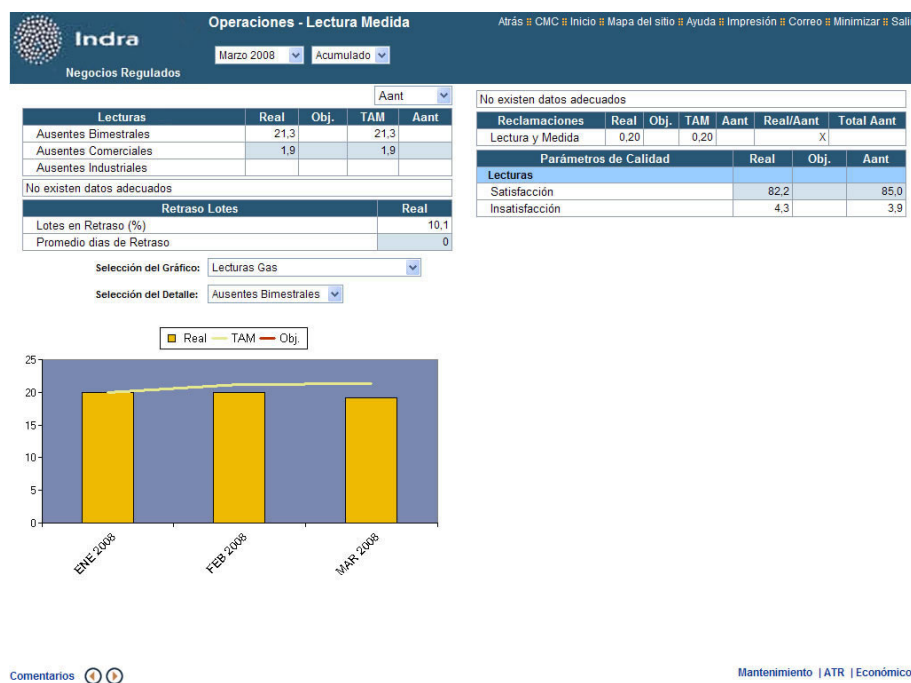
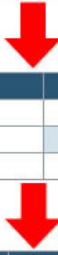


Figura 70: Pantalla Operaciones - Lectura y Medida

Como vemos en la imagen anterior, sobre la primera tabla, tenemos un selector que contiene “Aant”, que es el contenido de la última columna de la tabla. En las siguientes imágenes, podemos ver que al cambiar el valor de este selector, también cambia el nombre de la última columna de la cabecera y el contenido de toda la columna, tomando los valores de dicha característica.



				Aant
Lecturas	Real	Obj.	TAM	Aant
Ausentes Bimestrales	21,3		21,3	
Ausentes Comerciales	1,9		1,9	
Ausentes Industriales				

				Aant
Lecturas	Real	Obj.	TAM	Aant
Ausentes Bimestrales	21,3			Real/Aant
Ausentes Comerciales	1,9			Real/Prev
Ausentes Industriales				Total Aant

				Real/Prev
Lecturas	Real	Obj.	TAM	Real/Prev
Ausentes Bimestrales	21,3		21,3	X
Ausentes Comerciales	1,9		1,9	X
Ausentes Industriales				0,0

Figura 71: Cambio de columna visible en la tabla

Para conseguir este cambio, cuando definimos la tabla (en la función `iniPantalla()`), hemos añadido el comando `hideColumn` teniendo como parámetro el nombre de la tabla y la columna que queríamos que no apareciera por pantalla.

```
function iniPantalla()
{
  //TABLA1
  var tbl = getTableByName('TABLE_1');
  if (tbl.rows.length>1)
  {
    formateoFilaTabla(tbl,0,'tblCabecera');
    cambiaTextCelda(tbl,0,0,"Lecturas");
    hideColumn('TABLE_1',4);
    hideColumn('TABLE_1',5);
    hideColumn('TABLE_1',6);
    hideColumn('TABLE_1',7); }
  //...
}
```

Después en el body del HTML, declaramos el selector:

```
<span class='EnlacesMenuDiv'>
<select name="col1"
onchange="muestraRatio('col1', 'col1', '5', 'TABLE_1') " >
<option value=4>Aant</option>
<option value=5>Real/Aant</option>
<option value=6>Real/Prev</option>
<option value=7>Total Aant</option>
</select>
</span>
```

Debemos darle un nombre y los valores de las opciones deben ser el número de la columna de la tabla o tablas que queremos ocultar/mostrar.

Una vez realizado esto en el onchange del tag del select llamaremos a la función muestraRatio().

```
<select name="col1"
onchange="muestraRatio ('col1', 'col1', '5', 'TABLE_1') " >
```

Cuyos parámetros son:

- @nom_sel: nombre del selector que hemos declarado.
- @nom_cookie: nombre de la cookie donde guardaremos el valor escogido por el usuario.
- @valor_defecto: valor por defecto de la cookie, valor que aparece por primera vez en el selector y también es el número de la columna que mostrará cuando se cargue la página sin que el usuario haya cambiado el selector.
- @tables: nombre de la/s tabla/s que queremos que el selector afecte. Se pueden poner entre 1 y 9 tablas, la función ya se encarga de ver si la tabla tiene o no filas para saber si puede tratar la tabla o no.

En el onLoad del BODY o dentro de una función que sea llamada en el onLoad, debe llamarse a la función iniMuestraRatio(nom_sel, nom_cookie, valor_defecto, Tables), esta función tiene los mismos parámetros que muestraRatio, pero en vez de cambiar el valor de la cookie y ocultar una columna y mostrar otra, esta función carga el valor de la cookie, muestra la columna que corresponda al valor de esta y pone el selector con la opción de la columna de la cookie.

```
<BODY scroll="no" onLoad="inicializar();
iniMuestraRatio('col1','col1','5','TABLE_1'); ... " onUnload="salir()">
```


8. Conclusiones

Si hace un año me hubieran dicho que ahora presentaría un proyecto de fin de carrera con SAP, no me lo habría creído, pero una serie de hechos que fueron apareciendo a su debido tiempo me han llevado ante esta situación. Todo comenzó el verano pasado cuando la EPS en colaboración con Indra, ofrecieron un curso en tecnología SAP, desconocida para mi, pero que con el paso de los días, se fue convirtiendo en un interesante proyecto de futuro para mi vida profesional. Concluido el curso y tras las pertinentes rondas de entrevistas, la empresa me enroló en sus filas y así seguimos hasta ahora.

Una vez dentro del gigante nacional de la informática, me asignaron al área de inteligencia empresarial (SAP BW) y comenzamos este proyecto que va a marcar una parte de mí, pues significó el comienzo de mi vida laboral y supondrá el fin de mi época universitaria.

Durante este tiempo, hemos estado desarrollando esta aplicación junto a dos compañeros con una formación progresiva, puesto que al inicio del proyecto teníamos unos conocimientos básicos que han ido creciendo a medida que el proyecto iba evolucionando hasta llegar a su fin.

Una vez concluido el proyecto, podemos afirmar que el desarrollo y la implantación del cuadro de mando de Negocios Regulados de nuestro cliente ha sido un éxito. Esta rotunda afirmación se debe a que hemos ido cumpliendo los plazos de entrega y los problemas encontrados han sido solucionados a tiempo satisfactoriamente. El hecho que la aplicación esté funcionando correctamente, también es otro motivo para pensar que el trabajo se ha realizado cumpliendo los objetivos iniciales.

Pero como sucede siempre, no todo es un camino de rosas, y durante el desarrollo nos hemos encontrado con diversos problemas que gracias a la ayuda de los analistas o de invertir más horas hemos sabido superar y finalizar en las fechas establecidas. Estos problemas, generalmente han venido por nuestra poca experiencia, como todos sabemos nadie nace enseñado, pero con paciencia y esmero todo puede salir adelante. Esperemos que a partir de ahora, estas piedras en el camino, ya superadas, nos sirvan para recordar quienes fuimos y sepamos ayudar a los que vendrán y pasarán por la situación que vivimos meses atrás.

Y antes de que estas conclusiones tomen tintes literarios, veremos las mejoras que puede y debe afrontar esta aplicación y el trabajo que queda por hacer en las posibles ampliaciones que quiera ejercer nuestro cliente.

En el apartado de usabilidad, 1.5, hemos comentado que la rapidez del sistema no es la deseada en algunas ocasiones porque cada vez que hay una modificación en la pantalla, siempre se cargan en la memoria caché todas las Queries utilizadas en esa pantalla. Este tema, es una de las modificaciones que estamos intentando mejorar en estos momentos mediante una carga de las Queries a la caché cada cierto tiempo, ya que el cliente nos ha asignado el mantenimiento del proyecto una vez finalizado.

Para finalizar, comentar que este proyecto no acaba aquí su trayecto. Antes de empezar este cuadro de mando, la compañía que nos solicitó su desarrollo ya disponía de otro cuadro de mando para otra de sus áreas y posiblemente deseará llevar a cabo un cuadro de mando para cada una de sus áreas de responsabilidad, las cuales se apoyarán para las futuras tomas de decisiones en estas aplicaciones de inteligencia empresarial.

9.1. La ayuda de la aplicación

9.2. Funciones javascript utilizadas

Para el proyecto de CNR usamos Javascript para la creación de las URL's, las funciones con las que ocultamos o mostramos columnas, etc.

La mayoría de funciones, constantes o variables están declaradas en los archivos de Javascript del servidor de Sap, ubicados en:

Transacción SE80: Repository MIME ->SAP ->PUBLIC ->BW ->Customer ->JavaScript

Principales archivos JavaScript:

- ScrollableTable.js: Funciones referidas a las tablas de las pantallas, como por ejemplo las funciones de ocultar/mostrar columnas de tablas.
- funciones_propias_v8.js: Funciones de carácter general (todas las pantallas) para el filtrado de pantallas y la creación de las URLs.
- variables.js: Declaración de unas variables que son usadas como referencia para saber que tipo de dato sacar en cada mes para la columna de MPA.

En CM_CNR:

- CNR_contenidos.js: Aquí se ubican las funciones que se usan concretamente en CNR, pero que no son necesarias para otros cuadros de mandos.
- CNR_enlaces.js: En este archivo se declaran los aEnlaces que luego se usan para crear los links de pantallas, también se declaran los footlinks y las excepciones del carrusel.

A continuación mostraremos el código de los scripts que más hemos utilizado en el desarrollo de las pantallas.

Empezaremos con la función utilizada para cargar los ficheros JavaScript que utilizaremos en el Web Template.

```
function getFicherosJs()
{
var strAux;
strAux =
"<script type='text/javascript'
src='/sap/bw/Mime/Customer/JavaScript/funciones_propias_v8b.js'>
</script>"
+ "<script type='text/javascript'
src='/sap/bw/Mime/Customer/JavaScript/ScrollableTable.js'>
</script>"
+ "<script type='text/javascript'
src='/sap/bw/Mime/Customer/JavaScript/variables.js'>
</script>"
+ "<script type='text/javascript'
src='/sap/bw/Mime/Customer/JavaScript/CM_CNR/CNR_enlaces.js'>
</script>"
+ "<script type='text/javascript'
src='/sap/bw/Mime/Customer/JavaScript/CMC_CMGR_contenidos.js'>
</script>"
+ "<script type='text/javascript'
src='/sap/bw/Mime/Customer/JavaScript/CMC_jerarquias_manuales.js'>
</script>"
+ "<OBJECT ID='Func_Aux' "
+ "CLASSID='CLSID:5783AB8F-3C1F-48D7-BACE-9E310BBA6140' "
+ "CODEBASE='FuncAux.CAB#version=1,0,0,0'>"
+ "</OBJECT>"
+ "</script>"; document.write(strAux);
}
```

Con esta función, cambiaremos el texto de la tabla, la usaremos generalmente para titular las tablas que aparecen en la pantalla.

```
function cambiaTextCelda(tbl,iRowDest,iColDest,strTexto)
{
var headerrow ,celda, iPos;
if (typeof tbl == 'string') tbl = getTableByName(tbl);
headerrow = tbl.tBodies(0).rows(iRowDest);
headerrow.cells(iColDest).innerHTML=strTexto;
}
```

Los siguientes scripts los utilizamos para mostrar u ocultar columnas. Estos cambios de columnas son necesarios en las tablas que tenemos más columnas de las que podríamos mostrar por pantalla debido a las limitaciones del tamaño de la aplicación.

```
function mostrarColumna(tbl,iCol)
{
  var row ,celda, iPos;
  if (typeof tbl == 'string') tbl = getTableByName(tbl);
  //row = tbl.tBodies(0).rows(0);
  for (var i=0; i <tbl.tBodies(0).rows.length; i++) {
    row = tbl.tBodies(0).rows(i);
    row.style.display = "";
  }
}
```

```
function hideColumn(table, col)
{
  obj = getTableReference(table);
  if (obj == null)
  {
    alert("No Object Found");
  }
  else
  {
    for (var rowno =0;rowno <obj.rows.length;rowno++)
      obj.rows(rowno).cells(parseInt(col)).style.display = "none";
  }
}
```

```
function showColumn(table, col)
{
  obj = getTableReference(table);
  if (obj == null)
  {
    alert("No Object Found");
  }
  else
  {
    for (var rowno =0;rowno <obj.rows.length;rowno++)
      obj.rows(rowno).cells(parseInt(col)).style.display = "";
  }
}
```

Evento que guarda el estado de los scroll de los container's

```
function onScrollGuardaEstado()
{
for (var i=0; i < container.length; i++)
{
aScrollContainer[i] = container[i].scrollTop;
if (container[i].children.length>0)
{
aTblContainer[i] = container[i].children[0].name;
}
}
}
```

Funciones para utilizar selectores.

```
/** Construye las opciones de un select
 * @param arrOpciones: array con opciones:
new array(new array('cod1','desc1'),new array('cod2','desc2'),...)
 * @param strSeleccion: valor de la opción seleccionada por defecto */
function crearOpcionesSelect(arrOpciones, strSeleccion)
{
var i;
var aOpciones
var aAux;
var strRet;
strRet="";
for (i=0;i<arrOpciones.length;i++)
{
aAux = arrOpciones[i];
if (aAux[0] == strSeleccion)
{
strRet = strRet + "<OPTION value =' " + aAux[0] + " ' SELECTED>" +
aAux[1] + "</OPTION>";
}
else
{
strRet = strRet + "<OPTION value =' " + aAux[0] + " '>" + aAux[1] +
"</OPTION>";
}
}
return strRet;
}
```

```

/** Incluye un formulario para un select*/
function getSelForm (strNameForm, arrOpciones, strSelect,
strOnChange,strBloq)
{
var strAux;
if(strBloq == "1")
{
strAux = "<form name='"+ strNameForm + "'>" + " <SELECT
name='sSELECTOR1' class=SAPBexDdl size 1 onchange=" + strOnChange
+ " disabled>" + crearOpcionesSelect(arrOpciones,strSelect) +
" </SELECT>" + "</form>";
}
else
{
strAux = "<form name='"+ strNameForm + "'>" + "<SELECT
name='sSELECTOR1' class=SAPBexDdl size 1 onchange=" + strOnChange
+ ">" + crearOpcionesSelect(arrOpciones,strSelect) + "</SELECT>" +
"</form>";
}
return strAux;
}

```

```

/** Incluye en el documento las opciones de un select
* @param strOpciones: string con opciones: valor_1-texto_1, ...,
valor_n-texto_n
* @param strSeleccion: valor de la opción seleccionada por defecto */
function escribirOpcionesSelect(strOpciones, strSeleccion)
{
var strAux;
strAux = crearOpcionesSelect(strOpciones,strSeleccion);
document.write(strAux);
}

```


Funciones para la navegación de pantallas.

```
<!-- Scripts para secuencia de navegación -->
<!-- Devuelve el valor asignado en la URL al parametro que le pasamos -->
function getBWURLParam(parametro)
{
  var namevalue = SAP_BW_URL_Get().split('=');
  var valor;
  var nparam;
  for(i=0;i<namevalue.length;i++)
  {
    valor = namevalue[i].split('&');
    for(j=0;j<valor.length;j++)
    {
      if(nparam==parametro)
      {
        return valor[j];
      }
      nparam=valor[j];
    }
  }
}
```

```
<!-- Devuelve true si no es la primera vez que se accede a la página, es decir, si se está navegando dentro de la propia página -->
function continua_secuencia()
{
  var pagina = getBWURLParam('PAGENO');
  var pagina_anterior = getEstado('PAGINA');
  if (pagina_anterior)
  if (pagina == pagina_anterior)
  {
    var peticion = eval(getBWURLParam('REQUEST_NO'));
    var peticion_anterior = eval(getEstado('PETICION'));
    if (peticion > peticion_anterior ) return true;
  }
  else return false;
  return true;
}
```

```

<!-- Antes de salir de la pantalla establece las variables de estado: Pantalla
Actual y anterior, pagina y petición ->
function guarda_estado()
{
setEstado('PANTALLA_ACTUAL',SAPBWGetProperty('TEMPLATE_ID'));
// setEstado('PAGINA',getBWURLParam('PAGENO'));
// setEstado('PETICION',getBWURLParam('REQUEST_NO')); }

```

Funciones para actualización de variables.

```

<!-- Genera la cadena necesaria para establecer una variable vía URL ->
// Params: (string) varname1 = variable a establecer (string)
// (string) valor1 = valor a asignar.
function getURLVar(varname1,valor1)
{
var cadena = "&CMD=PROCESS_VARIABLES&SUBCMD =
VAR_SUBMIT&"; if (varname1 != "")
return (cadena + "VAR_NAME_1 = "
+varname1+"&VAR_VALUE_EXT_1 = "+valor1);

```

Funciones que utilizamos para tratar las cookies.

```

var cm_cookie = "";
var nomCookie = 'cCMGN';
<!-- Guarda en una Cookie una variable y su valor -->
function setEstado(nombre,valor)
{
var sValor;
var ini, fin;
var re = new RegExp(nomCookie + "([^;]+)");
var cm_cookie_reg = re.exec(document.cookie);
if (valor)
sValor = nombre + ":" + valor + '/';
else
sValor = nombre + ":nulo/";
if (cm_cookie_reg != null)
{
cm_cookie = cm_cookie_reg[1];
ini = cm_cookie.indexOf(nombre + ':');
fin = cm_cookie.indexOf("/",ini);
if (fin == -1)
fin = cm_cookie.length;
if (ini == -1)
cm_cookie = cm_cookie + sValor;
else
cm_cookie = cm_cookie.substring(0,ini) + sValor +
cm_cookie.substring(fin+1,cm_cookie.length);
}
else
cm_cookie = sValor;
document.cookie = nomCookie + '=' + cm_cookie;
return valor; }

```

```

<!-- Recupera de una Cookie el valor de la variable que se le pasa como parámetro -->
->
function getEstado(nombre,valor_defecto)
{
var valor;
var ini = document.cookie.indexOf(nomCookie+'=');
if(ini == -1) // no existe ese valor
if (valor_defecto)
return valor_defecto
else return ""
var sep = document.cookie.indexOf("=", ini);
var fin = document.cookie.indexOf(";", ini);
if(fin == -1) // el último no acaba en ;
fin = document.cookie.length;
cm_cookie = document.cookie.substring(sep+1, fin);
var exre = new RegExp(nombre + ":([/]+)");
var valor_reg = exre.exec(cm_cookie);
if (valor_reg!=null) valor = valor_reg[1];
else valor = valor_defecto
if (valor=='nulo') return ""
else return valor; }

```

```

<!-- Establece valores en las variables almacenadas en la Cookie de la sesión -->
// Params: (string) variables = Lista separada por comas de los nombres de las
variables a establecer Ej: 'moneda,participacion'.
// (string) valores = Lista separada por comas de los valores de las variables.
Ej: 'EU,P'.
// Devuelve: blanco.
function setValorCookies(variables,valores)
{
var vVariables = (variables.indexOf(",")>-1)?
variables.split(','):new Array(variables);
var vValores = (valores.indexOf(",")>-1)?
valores.split(','):new Array(valores);
for (var i=0; i<vVariables.length;i++)
{
setEstado(vVariables[i],vValores[i]);
}
return "";
}

```

```

<!-- Recupera el valor para la variable mensual -->
function getMensual()
{
    var iAux = getAcumulado();
    if (iAux==1) return 0;
    else return 1;
}

```

```

<!-- Devuelve el código de la pantalla visitada con anterioridad -->
function getPantallaAnterior()
{
    return getEstado('PANTALLA_ ANTERIOR');
}

```

Otras funciones útiles.

```

// Genera el filtro para la lista de infoobjetos que recibe como parámetro. Si
// alguno de ellos debe ser un filtro de exclusión se pasa con el sufijo .E
function getURLFilter(fiobjnms,fvalues,extension)
{
    var iobjetos, valores;
    var iobj;
    var cadena = "";
    iobjetos = (fiobjnms.indexOf(",")>-1)? fiobjnms.split(','):new Array(fiobjnms)
    if (fvalues)
        valores = (fvalues.indexOf(",")>-1)? fvalues.split(','):new Array(fvalues);
    else
    {
        valores = new Array(); // Recuperar de la cookie
        for (var i=0;i<iobjetos.length;i++)
            valores[i]=getEstado(iobjetos[i].split('.')[0])
    }
    for (var i=0;i<iobjetos.length;i++)
    {
        iobj = iobjetos[i].split('.');
        if(getEstado(iobj[0]))
        {
            cadena += '&FILTER_I OBJNM_' + (i+1) + '=' + iobj[0] +
            '&FILTER_ VALUE_' + (i+1) + '=' + valores[i];
            if(iobj[1]) cadena += '&FILTER_ SIGN_' + (i+1) + '=' + iobj[1];
        }
    }
    if (extension) cadena += extension; return cadena;
}

```

Manejo de elementos BW

```
<!-- Retorna un objeto el primer objeto de tipo tabla con el nombre que se le
pasa como parámetro -->
// Pre: La tabla buscada debe existir.
// Params: (string) tabla: Nombre de la tabla buscada (Nombre que se le da al
web item Tabla)
function getTableByName(tabla)
{
  if (typeof tabla != 'string') return tabla;
  var tempEle = document.getElementsByTagName('table');
  for(var i=0;i<tempEle.length;i++)
  {
    if((tempEle[i].name==tabla) && (tempEle[i].id!="Header"))
      return tempEle[i];
  }
}
```

```
function getSelectByName(select)
{
  if (typeof select != 'string') return select;
  var tempEle = document.getElementsByTagName('select');
  for(var i=0;i<tempEle.length;i++)
  {
    if(tempEle[i].name==select) return tempEle[i];
  }
}
```

```
<!-- Retorna el primer objeto formulario asociado al web item que se le pasa
como parámetro -->
// Pre: El web item debe existir y debe estar dentro de un formulario.
// Params: (string) nombre: Nombre del web item buscado.
function getBWObjectByName(nombre)
{
  if (typeof nombre != 'string') return nombre;
  var tempEle;
  tempEle = document.getElementsByTagName("ITEM");
  for(var i=0;i<tempEle.length;i++)
    if(tempEle[i].value==nombre) return tempEle[i].parentNode;
}
```

Función utilizada para formatear con otra clase una línea de una tabla.

```

//Incluida la variable sinAlerta para evitar la visualización de alertas si las
hubiese
//Opción de formatear fila asociando texto de celda (sin espacios) o número de
fila según el tipo de la variable iOpcion
function formateoFilaTabla(strTable, iOpcion, strClase, iColRight, sinAlerta)
{
var tbl;
var espacios=new RegExp(" ","g");
var irow;
var iPosOtra;
if (iColRight)
{
iPosOtra= iColRight;
}
else
{
iPosOtra = 9999;
}
tbl = getTableByName(strTable);
//si pasamos número de fila
if(typeof(iOpcion)=="number")
{ if (tbl.rows.length>=iOpcion) irow = iOpcion; }
//si pasamos texto de celda
else if(typeof(iOpcion)=="string")
{
for(var i=0;i<tbl.rows.length;i++)
{
if(tbl.rows[i].cells[0].innerText.replace(espacios,"").toUpperCase() ==
iOpcion.toUpperCase()) irow = i;
}
}
if((irow)||(irow==0))
{
//si el string iOpcion no está visible debido al desplegable no hago nada
for (var i=0;i<tbl.rows(irow).cells.length;i++)
{
strClaseAnt = tbl.rows(irow).cells(i).className;
var iPos = strClaseAnt.lastIndexOf("Critical");
var iPos2 = strClaseAnt.lastIndexOf("Bad");
var iPos3 = strClaseAnt.lastIndexOf("Good");
if ((sinAlerta == true) || ((iPos== -1) && (iPos2== -1) && (iPos3== -1)))
{
if (i<iPosOtra) { tbl.rows(irow).cells(i).className=strClase;
}
else
{
tbl.rows(irow).cells(i).className= strClase + "_R";
}
}
}
}
}
}
}
}
}

```

10. Bibliografía

Manuales utilizados:

- SAP AG. “BW310 Data Warehousing”. 2006
- SAP AG. “BC400 Introduction to the ABAP Workbench”. 2006

Páginas Web consultadas:

- Wikipedia
<http://es.wikipedia.org/>
- Mundo SAP
<http://www.mundosap.com/>
- Ayuda de SAP
<http://help.sap.com/>
- Ayuda de SAP Business Information Warehouse
http://help.sap.com/saphelp_nw04/helpdata/en/a8/6b023b6069d22ee10000000a11402f/frameset.htm